

2007/87



Valid inequalities for the single-item capacitated lot
sizing problem with step-wise costs

Ayşe Akbalık and Yves Pochet

CORE DISCUSSION PAPER
2007/87

**Valid inequalities for the single-item capacitated
lot sizing problem with step-wise costs**

Ayşe AKBALIK¹ and Yves POCHET²

November 2007

Abstract

This paper presents a new class of valid inequalities for the single-item capacitated lot sizing problem with step-wise production costs (LS-SW). We first provide a survey of different optimization methods proposed to solve LS-SW. Then, *flow cover* and *flow cover inequalities* derived from the single node flow set are described in order to generate the new class of valid inequalities. The single node flow set can be seen as a generalization of some valid relaxations of LS-SW. A new class of valid inequalities we call *mixed flow cover*, is derived from the integer flow cover inequalities by a lifting procedure. The lifting coefficients are sequence independent when the batch sizes (V) and the production capacities (P) are constant and if V divides P . When the restriction of the divisibility is removed, the lifting coefficients are shown to be sequence independent. We identify some cases where the mixed flow cover inequalities are facet defining. A cutting plane algorithm is proposed for these three classes of valid inequalities. The exact separation algorithm proposed for the constant capacitated case runs in polynomial time. Finally, some computational results are given to compare the performance of the different optimization methods including the new class of valid inequalities.

Keywords: single-item capacitated lot sizing problem, flow cover inequalities, cutting plane algorithm.

¹ CORE, Université catholique de Louvain, Belgium. E-mail: ayse.akbalik-rapine@cea.fr

² CORE, Université catholique de Louvain, Belgium. E-mail: yves.pochet@lhoist.com

Work carried out when A. Akbalik was Post-Doctoral Fellow in 2006-2007 at CORE, Université catholique de Louvain.

This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1 Introduction

In this paper, we focus on a single-item capacitated lot sizing problem (LSP) having a special cost structure. The production cost is assumed to be step-wise (piece-wise linear with discontinuous steps), where each step has a length of the batch size produced in each period. We denote the problem by LS-SW. The methods proposed for some restrictive cases of the classical LSP can no more be used to solve LS-SW because of its cost structure. The attractiveness of this simple LS-SW case lies in the need to solve more complex production and inventory problems where LS-SW arises as a submodel. To better understand and to solve efficiently the single-item LSP can then be helpful to solve more complex production planning problems. In this paper, we give a detailed survey on different optimization methods of LS-SW, followed by the description of a new class of valid inequalities which constitutes the main contribution.

The reader can refer to Pochet and Wolsey [16] for a comprehensive literature survey, particularly for a detailed information on the mixed integer programming approach to solve production planning problems. Another detailed review on the single-item LSP can be found in Brahimi *et al.* [3]. One of the first studies on the integration of the fixed cost per batch (or fixed transportation costs) into the inventory control policy is from Lippman [13]. Transportation cost is assumed to be concave and step-wise. Under some additional hypotheses on the holding and production costs, the author gives the properties of an optimal solution. Swoveland [20] considers a single product, multi-period production planning model where production and holding-backorder cost functions are assumed to be piece-wise concave. Some optimal schedule properties and a dynamic programming algorithm are given.

Lee *et al.* [11] study an integrated inventory replenishment and outbound dispatch scheduling problem. They adopt a network approach to solve it and propose polynomial time algorithms using some optimal solution properties. Li *et al.* [12] study two variants of the dynamic economic lot sizing problem. In the first one, the production is restricted to be a multiple of a fixed batch size in each period, and the costs are time varying, whereas in the second, a more general form of product order cost structure is assumed. Polynomial time algorithms are proposed for each case. Jin and Muriel [10] study a system composed of one warehouse receiving a single product from a supplier and replenishing the inventory of n retailers with direct shipments. The process of ordering from the supplier and shipping to the retailers generates full truckload transportation costs with cargo capacity constraints. Giving some optimal solution properties, they study decentralized and centralized systems.

There are some studies on the mixed integer linear programming (MILP) approach to solve the piece-wise linear cost LSP. Diaby and Martel [7] study an arborescent, multi-echelon distribution system to determine optimal purchasing and shipping quantities. They consider general piecewise linear procurement cost and linear holding costs. They formulate the problem as a MILP and propose a lagrangean relaxation. Chan *et al.* [5] study the less-than-truckload shipments problem integrated with the production and inventory activities. The cost function is piece-wise linear and concave. They model the problem as a concave cost multi-commodity network flow problem. They formulate it as a MILP using a set-partitioning approach, and characterize structural properties. Chan *et al.* [4] study a special class of piece-wise linear ordering cost LSP, which they refer to as *modified all-unit discount* cost function. This function arises when the transportation is done with less-than-truckload carriers. They prove this problem to be NP-hard and construct the best zero inventory ordering policy and then compare it with the optimal policy.

The main contribution of this paper is a new class of valid inequalities for LS-SW. These inequalities are derived for some relaxations of LS-SW from the integer flow cover inequality (IFCI), which is a slight variation of the flow cover inequalities (FCI). Flow and integer flow cover inequalities exist in the literature and are derived from the single node flow set which is more general than the relaxed subparts of LS-SW. First studies on the flow cover inequalities are from Padberg *et al.* [15]. The integer flow cover inequalities are first studied by Atamtürk [2]. For

more detailed information on the flow cover inequalities, see section 4.

The LS-SW problem description and a mathematical formulation are given in Section 2. Some optimization methods proposed in the literature are described in Section 3. Two classes of valid inequalities existing in the literature and a new one are shown to be valid (and in some cases facet defining) for different relaxations of LS-SW in Section 4. In Section 5, a cutting plane algorithm is proposed to use these three classes of VI's in a Branch&Cut procedure. A polynomial time separation algorithm is described in the same section. Finally, some computational results and concluding remarks are presented in Sections 6 and 7.

2 Problem description

Some real applications of the LS-SW can be cited as follows. Different batch sizes (V_t) of a single-item are produced in a given period t , in a capacitated plant where one period capacity is limited to P_t . The finished products are stored at the end of each period in the plant. There is a setup cost for producing a positive quantity and also a fixed cost per batch produced. The aim is to satisfy the customer demand, which is known on a given finite horizon, while minimizing the total production and inventory costs. Another application can be imagined as a capacitated plant (P_t) producing a single-item and sending the finished products to a distribution center (DC) with capacitated vehicles (V_t). In the plant there is no storage space and the final demand of the customer is known in DC. A fixed transportation cost is associated with each vehicle shipped and again a setup cost is generated for any positive quantity produced.

For the two cases described above, the cost structure for the production activity becomes step-wise because of the fixed costs per batch (or per vehicle). The mathematical formulation by MILP of LS-SW is given as follows. We denote by X^{LS-SW} the set of points (x, y, z, s) satisfying constraints (1) to (7).

Data P_t : production capacity in period t , V_t : batch size (or vehicle capacity) in period t , q_t : setup cost in period t , p_t : unit production cost in period t , f_t : fixed cost per batch (per vehicle) in period t , h_t : unit holding cost at the end of period t , d_t : demand in period t , n : number of periods on the planning horizon.

Decision variables x_t : production quantity in period t , y_t : binary setup variable in period t , z_t : number of batches (or vehicles) in period t , s_t : storage quantity at the end of period t .

$$\min \sum_{t=1}^n (q_t y_t + p_t x_t + f_t z_t) + \sum_{t=0}^n h_t s_t$$

s.t.

$$s_{t-1} + x_t = s_t + d_t, \forall t \in \{1, \dots, n\} \quad (1)$$

$$x_t \leq P_t y_t, \forall t \in \{1, \dots, n\} \quad (2)$$

$$x_t \leq V_t z_t, \forall t \in \{1, \dots, n\} \quad (3)$$

$$y_t \leq z_t, \forall t \in \{1, \dots, n\} \quad (4)$$

$$z_t \leq \lceil \frac{P_t}{V_t} \rceil y_t, \forall t \in \{1, \dots, n\} \quad (5)$$

$$s_0 = s_n = 0 \quad (6)$$

$$x \in \mathbb{R}_+^n, s \in \mathbb{R}_+^{n+1}, y \in \{0, 1\}^n, z \in \mathbb{Z}_+^n \quad (7)$$

The most natural formulation of the problem is given above. In the literature this formulation is denoted by AGG (aggregated formulation). The objective function minimizes the total cost of production, storage and transportation. Constraint (1) represents the material balance in each

period. The restriction on the production capacity and the generation of a setup variable are expressed by constraint (2). In constraint (3), the number of batches (or vehicles) is computed as a function of the quantity produced in each period. Constraint (4) forces the number of batches (or vehicles) to be positive if a setup is done at this period. Constraint (5) expresses the limitation on the number of batches (or vehicles) and forces it to be 0 if there is no production. Constraint (6) is an assumption on the storage quantities at the beginning of the first and at the end of the final period. In constraint (7), the feasibility domain of each decision variable is given.

Florian *et al.* [8] give an NP-hardness proof for the single-item capacitated LSP with equal demands and zero storage costs. LS-SW is more general than this case, which implies its NP-hardness. There exists an exact pseudo-polynomial time dynamic programming algorithm to solve LS-SW in the general case (see Shaw and Wagelmans [19]). Therefore, this one is NP-hard in the ordinary sense.

3 Optimization methods

The approach mostly used for the single-item lot sizing problem is dynamic programming. Moreover, it is possible to find some polynomial cases using this approach. However, for the more complex extensions (multi-item, backlogging, time window shipments, etc.), it becomes inefficient. In section 3.1, we give the main ideas of different recursive formulations for LS-SW. Another approach is to find some subsets of X^{LS-SW} defining valid relaxations for this set and for which there exists a complete description of the convex hull. We classify them into two axes: uncapacitated relaxations and capacitated relaxations. Formulations for these relaxations are implemented in LS-LIB¹.

3.1 Dynamic programming approach

For the general case when the parameters can take arbitrary values, the recursive formulation by Florian *et al.* [8] can be used. The only change made for LS-SW compared to the classical case is the total cost computation which does not change the complexity. This formulation has a time complexity in $O(T^2 \bar{P} \bar{d})$ with \bar{P} , the average production capacity over horizon and \bar{d} , the average demand. Another formulation for the general case is from Chen *et al.* [6], which has a time complexity in $O(T^3 \bar{d}^2)$.

Shaw and Wagelmans [19] provide a pseudo-polynomial time algorithm with a complexity in $O(T^2 \bar{q} \bar{d})$ for the constant capacitated LSP with piece-wise linear costs with \bar{q} , the number of pieces in the production cost function and \bar{d} , the average demand. For the classical case where the production cost has only one setup component, the complexity becomes $O(T^2 \bar{d})$ with $\bar{q} = 1$. This is a significant improvement compared to the pseudo-polynomial dynamic programming algorithm proposed by Florian *et al.* [8]. Finally, one can cite the studies from Akbalık and Rapine [1], where the authors give three polynomial cases for LS-SW under some restrictive hypotheses on the capacities and costs of the production and of the vehicles.

3.2 Mixed integer linear programming reformulations

For two special cases of the single-item capacitated lot sizing problem, Pochet and Wolsey [17] propose extended formulations where matrices are totally unimodular. For the first case, the production capacity is assumed to be constant and a setup cost is paid for each positive amount produced. For the second case, the production capacity in each period is assumed to be an integer which is multiple of some batch size and a fixed cost is generated for each batch produced. However, for the special case of LS-SW with the assumptions of constant capacity production

¹<http://www.core.ucl.ac.be/LS-LIB/PPbyMIP>. A library of reformulations, cutting planes and heuristics. Both Mosel and Xpress-MP are required in order to use LS-LIB.

($P_t = P, \forall t$), constant sized batch production ($V_t = V, \forall t$) and fixed costs per batch, we could not manage to find an extended formulation whose linear relaxation gives an optimal solution.

3.2.1 Uncapacitated relaxations

There are 4 known compact formulations for the uncapacitated lot sizing problem (ULSP), see Pochet and Wolsey [16]. To obtain the classical ULSP, one has to relax transportation and production capacity constraints from the formulation of LS-SW (constraints (3), (4) and (5) are relaxed, while in constraint (2) P_t is replaced by $\sum_{i=t}^n d_i$). For the linear relaxation of ULSP, these four formulations give integer optimal solutions.

The first one is based on the facility location problem formulation (**FAL**). Decision variables x_q are decomposed into x_{qt} which is the quantity produced in period q to satisfy the demand in period t . Decision variables of the second formulation and the idea of modeling is similar to FAL. It is based on the multi-commodity formulation (**MC**) where each demand in a given period is considered as a separate commodity. The third one is based on the shortest path formulation (**SP**) which is deduced from the dynamic programming recursion solving uncapacitated LSP. These 3 formulations loose their property of integrality with the addition of production and transportation capacity restrictions.

For the last uncapacitated relaxation that we denote by X^{WW} , one has to make an additional hypothesis on the costs, which is called Wagner-Whitin (WW) cost structure. For WW cost structure, producing and storing one unit in period t costs more than producing it later ($p_t + h_t \geq p_{t+1}, \forall t$). To obtain X^{WW} , constraint (1) is summed up for the interval $[t, k]$, $1 \leq t \leq k \leq n$ and s_k is eliminated with the assumption of $s_k \geq 0$. Constraints (2) to (5) are relaxed.

$$X^{WW} = \{(s, y) \in \mathbb{R}_+^{n+1} \times \{0, 1\}^n : s_{t-1} + \sum_{j=t}^k x_j \geq \sum_{j=t}^k d_j, 1 \leq t \leq k \leq n\}$$

Under WW cost structure, Pochet and Wolsey [18] give the convex hull for X^{WW} using the stock minimal solution (for which it is optimal to produce as late as possible). The complete description of the convex hull is given by the set X^{WW-U} . In total, one needs $O(T^2) \times O(T)$ constraints and variables for this description.

$$X^{WW-U} = \{(s, y) \in \mathbb{R}_+^{n+1} \times \{0, 1\}^n : s_{t-1} + \sum_{j=t}^k (d_{jk})y_j \geq \sum_{j=t}^k d_j, 1 \leq t \leq k \leq n\}$$

3.2.2 Constant capacitated relaxations

Two constant capacitated relaxations are given with their complete description.

Mixing Set Valid relaxations X^{WW-P} and X^{WW-V} , respectively for the cases where the production and vehicle capacities are constant, are described by (8) and (9). In a similar way as described above, we sum up constraint (1) for each interval $[t, k]$ with $1 \leq t \leq k \leq n$ and replace x_j by their upper bounds Py_j or Vz_j . Constraints (3), (4) and (5) are relaxed.

$$X^{WW-P} = \{(s, y) \in \mathbb{R}_+^{n+1} \times \{0, 1\}^n : s_{t-1} + \sum_{j=t}^k Py_j \geq d_{tk}, 1 \leq t \leq k \leq n\} \quad (8)$$

$$X^{WW-V} = \{(s, z) \in \mathbb{R}_+^{n+1} \times \mathbb{Z}_+^n : s_{t-1} + \sum_{j=t}^k Vz_j \geq d_{tk}, 1 \leq t \leq k \leq n\} \quad (9)$$

Under WW cost structure, the polyhedral characterization, the extended formulation and the separation algorithm for X^{WW-P} and X^{WW-V} can be found in Pochet and Wolsey [18].

Günlük and Pochet [9] studied the mixing set X^{MIX} described as follows,

$$X^{MIXING} = \{(s, y) \in \mathbb{R}_+^1 \times \mathbb{Z}^K : s + y_k \geq b_k \text{ for } 1 \leq k \leq K\}$$

For a given and fixed period t , X^{WW-P} and X^{WW-V} are equivalent to X^{MIXING} . A complete description of the convex hull of X^{MIXING} is known. The extended formulation for its convex hull exists with $O(T^2) \times O(T^2)$ constraints and variables. It is obtained by taking the convex hull of the extreme points and non-negative multiples of the extreme rays of the set X^{MIXING} .

Another important result concerning the convex hulls of X^{WW-P} and X^{WW-V} can be found in Pochet and Wolsey [16]. We mention the result only for X^{WW-P} , which remains the same for X^{WW-V} . For a given t , one can define the following set,

$$X_t^{M-P} = \{(s, y) \in \mathbb{R}_+ \times \{0, 1\}^n : s_{t-1} + \sum_{j=t}^k P y_j \geq d_{tk}, t \leq k \leq n\}$$

Therefore, $X^{WW-P} = \bigcap_t X_t^{M-P}$. The result is as follows,

$$\text{conv}(X^{WW-P}) = \text{conv}\left(\bigcap_t X_t^{M-P}\right) = \bigcap_t (\text{conv}(X_t^{M-P}))$$

Therefore, it suffices to take the intersection of the extended compact linear formulation for X_t^{M-P} to obtain a convex hull description of X^{WW-P} .

Divisible Mixing Set The last valid relaxation we consider for the constant capacitated LS-SW concerns the case in which the production quantity is a multiple of a constant batch size. This subset is denoted by X^{DM} . Constraint (1) is transformed as in previous sections but this time two separate inequalities are obtained by replacing $x_j \leftarrow P y_j$ and $x_j \leftarrow V z_j$ in the same set. Then, both inequalities are normalized dividing by V . Constraints (4) and (5) are also relaxed.

$$X^{DM} = \{(s, y, z) \in \mathbb{R}_+^{n+1} \times \{0, 1\}^n \times \mathbb{Z}_+^n : \frac{s_{t-1}}{V} + \sum_{j=t}^k z_j \geq \frac{\sum_{j=t}^k d_j}{V},$$

$$\frac{s_{t-1}}{V} + \frac{P}{V} \sum_{j=t}^k y_j \geq \frac{\sum_{j=t}^k d_j}{V}, 1 \leq t \leq k \leq n\}$$

Van Vyve [21] studied the divisible mixing set X^{DMIX} below,

$$X^{DMIX} = \{(s, y, z) \in \mathbb{R}_+^1 \times \mathbb{Z}_+^m \times \mathbb{Z}_+^m : s + y_k \geq b_k \text{ for } k \in K_1, s + C y_k \geq b_k, \text{ for } k \in K_2\}$$

For a given and fixed period t , X^{DM} is equivalent to X^{DMIX} , in which, $K_1 = K_2 = \{t, \dots, n\}$ and

$$s \leftarrow \frac{s_{t-1}}{V}, y_k \leftarrow \sum_{j=t}^k z_j \text{ for } k \in K_1, y_k \leftarrow \sum_{j=t}^k y_j \text{ for } k \in K_2,$$

$$b_k \leftarrow \frac{\sum_{j=t}^k d_j}{V} \text{ for } k \in K_1 \cup K_2, C \leftarrow \frac{P}{V}$$

An integral extended formulation for X^{DMIX} is proposed by Van Vyve [21] by taking the convex hull of the extreme points and non-negative multiples of the extreme rays of the set X^{DMIX} . $\text{Conv}(X^{DMIX})$ is also described in the original space of variables. The extended formulation is described by $O(T^3) \times O(T^2)$ constraints and variables. Therefore, for a given and fixed period t , the complete description of X^{DM} is known.

4 Valid inequalities

All results of the reformulation and relaxation explained in the previous sections exist in the literature and they have been proposed for the single-item lot sizing problem with constant capacity batch size (see Pochet and Wolsey [16]). We found some good relaxations of X^{LS-SW} which enabled us to use these reformulation results to solve LS-SW more efficiently. From now on, our aim is to find new valid inequalities for cases with varying capacities. Several classes of valid inequalities for the lot sizing problem with varying capacities are given in Pochet and Wolsey [16].

Analyzing different facet defining inequalities for X^{LS-SW} , we found two classes of valid inequalities existing in the literature, namely *flow cover inequalities (FCI)* and *integer flow cover inequalities (IFCI)*. They are derived from the single-node flow set which is a generalization of some valid relaxations of LS-SW. We propose a new class which is obtained by lifting the IFCI. For more details on lifting, the reader can refer to Louveaux and Wolsey [14]. Flow cover inequalities are described in this section after a motivating example which introduces the main idea of these inequalities and of the lifting procedure.

4.1 A first illustration

The first step is to find a good valid relaxation for the initial set X^{LS-SW} . A good valid relaxation means a subset of X^{LS-SW} for which valid or facet defining inequalities can be derived in order to use them to cut infeasible points in the initial formulation of X^{LS-SW} . One can construct a valid relaxation of the set X^{LS-SW} summing up constraint (1) for an interval $[t, k]$ with $1 \leq t \leq k \leq n$ and considering $s_{t-1} \geq 0$. Other constraints are all kept. Example below is constructed for $n = 3$, $d = (0, 0, 26)$, $P = 9$, $V = 3$, $t = 1$, $k = 3$ and assuming $s_0 = s_3 = 0$. Let X^e ,

$$X^e = \{(x, y, z) \in \mathbb{R}_+^3 \times \{0, 1\}^3 \times \mathbb{Z}_+^3 : x_1 + x_2 + x_3 \leq 26, x_i \leq 9y_i, x_i \leq 3z_i, y_i \leq z_i \leq 3y_i\}$$

The basic idea of the flow cover inequalities is to make constraint (10) stronger.

$$x_1 + x_2 + x_3 \leq 26 \tag{10}$$

The maximal set of affinely independent feasible integer points in X^e which make constraint (10) tight are as follows: $Set_1 = \{(9, 9, 8, 1, 1, 1, 3, 3, 3), (9, 8, 9, 1, 1, 1, 3, 3, 3), (8, 9, 9, 1, 1, 1, 3, 3, 3)\}$. One can observe that for these 3 points, the valid inequality $y_i \leq 1$ is also tight.

- The idea of the flow cover inequalities is to add the slack variables $(1 - y_i)$ with a certain coefficient ϕ_i in the left hand side of constraint (10). Let us first add $\phi_1(1 - y_1)$.

$$x_1 + x_2 + x_3 + \phi_1(1 - y_1) \leq 26$$

What is the maximum value of ϕ_1 to have the strongest inequality, without violating the initial constraints of the set X^e ? The procedure of lifting answers this question.

$$\phi_1 = 26 - \max\{x_1 + x_2 + x_3 : y_1 = 0, (x, y, z) \in X^e\} = 26 - \{0 + 9 + 9\} = 8$$

The same procedure can be applied to the other variables ($\phi_2 = \phi_3 = 8$) but this time with the addition of $\sum_{i=1}^{j-1} (\phi_i(1 - y_i))$. For instance,

$$\phi_3 = 26 - \max\{x_1 + x_2 + x_3 + 8(1 - y_1) + 8(1 - y_2) : y_3 = 0, (x, y, z) \in X^e\} = 8$$

ϕ_i are called the lifting coefficients since they lift the corresponding face to a higher dimension. In this example, the lifting sequence of the variables is not important. One can easily observe that constraint (10) is not tight for the following three points: $Set_2 =$

$\{(9, 9, 0, 1, 1, 0, 3, 3, 0), (9, 0, 9, 1, 0, 1, 3, 0, 3), (0, 9, 9, 0, 1, 1, 0, 3, 3)\}$. A stronger inequality (11) can thus be obtained which becomes tight for the previous 6 points (Set_1 and Set_2). (11) is called a flow cover inequality.

$$x_1 + x_2 + x_3 + 8(1 - y_1) + 8(1 - y_2) + 8(1 - y_3) \leq 26 \quad (11)$$

- The idea of the integer flow cover inequalities is to add the slack variables $(\lceil \frac{P}{V} \rceil - z_i)$ with a certain coefficient φ_i in the left hand side of constraint (10). Constraint (10) is not tight for the three following points:
 $Set_3 = \{(9, 9, 6, 1, 1, 1, 3, 3, 2), (9, 6, 9, 1, 1, 1, 3, 2, 3), (6, 9, 9, 1, 1, 1, 2, 3, 3)\}$. Let us first add $\varphi_1(3 - z_1)$ to the left hand side of constraint (10).

$$x_1 + x_2 + x_3 + \varphi_1(3 - z_1) \leq 26$$

Again, what is the maximum value of φ_1 to have the strongest inequality, without violating the initial constraints of the formulation? We tried all different values of z_i ($z_i = 0, 1, 2$ or 3) and chose the minimum value of φ_1 found. We only give the computation for $z_i = 2$.

$$\varphi_1 = 26 - \max\{x_1 + x_2 + x_3 : z_1 = 2, (x, y, z) \in X^e\} = 26 - \{6 + 9 + 9\} = 2$$

The same procedure can be applied to the other variables. All φ_i values are the same for all i , which shows that this computation is also sequence independent ($\varphi_2 = \varphi_3 = 2$). A stronger inequality (12) is thus obtained which becomes tight for the 6 points in Set_1 and Set_3 . (12) is called an integer flow cover inequality. Observe that inequality (12) is not tight for the three points in Set_2 .

$$x_1 + x_2 + x_3 + 2(3 - z_1) + 2(3 - z_2) + 2(3 - z_3) \leq 26 \quad (12)$$

- Finally, we show the idea of the new class of valid inequalities we propose in this paper. The dimension of the set X^e is 9. In order to obtain a facet defining inequality for the set X^e , one has to find 9 affinely independent points which satisfy the inequality as equality. Let us take the integer flow cover inequality and try to lift in slack variables $\phi'_i(1 - y_i)$.

$$x_1 + x_2 + x_3 + 2(3 - z_1) + 2(3 - z_2) + 2(3 - z_3) + \phi'_1(1 - y_1) \leq 26$$

$$\begin{aligned} \phi'_1 &= 26 - \max\{x_1 + x_2 + x_3 + 2(3 - z_1) + 2(3 - z_2) + 2(3 - z_3) : y_1 = 0, (x, y, z) \in X^e\} \\ &\Rightarrow \phi'_1 = 26 - \{0 + 9 + 9 + 6\} = 2 \end{aligned}$$

Using the same arguments, ϕ'_2 and ϕ'_3 are also found equal to 2 which shows that this procedure is also sequence independent. The valid inequality for X^e is thus,

$$x_1 + x_2 + x_3 - 2z_1 - 2z_2 - 2z_3 - 2y_1 - 2y_2 - 2y_3 \leq 2 \quad (13)$$

Inequality (13) is valid since the computation procedure of each coefficient satisfies all the initial constraints of X^e . 9 points in Set_1 , Set_2 and Set_3 satisfy inequality (13) as equality and are affinely independent, which makes this inequality facet defining for the set X^e .

4.2 Existing flow cover inequalities for LS-SW

Flow cover inequalities are derived from the polytope of the single node flow set. First studies in this area are from Padberg *et al.* [15]. The 0-1 single node flow set can be seen as a generalization of 0-1 knapsack sets or as a subproblem of a fixed charge network flow problem. For a given node of the network, this set represents the flow conservation constraint in this node.

We give a first valid relaxation for X^{LS-SW} and derive associated flow cover inequalities. The second valid relaxation allows us to use the known integer flow cover inequalities firstly introduced by Atamtürk [2]. The third class of valid inequalities constitutes a new class which is the main contribution of this paper.

4.2.1 Flow cover for a valid relaxation of X^{LS-SW}

The first valid relaxation of X^{LS-SW} considered corresponds to the single node flow set with only entering flows in the node. We denote it by X^{R1} . Each entering flow represents the quantity produced at a given period. The leaving arcs are the total demand in the corresponding time interval and the final stock. The first constraint represents the flow conservation in a node of the network. There are also associated binary variables (setup variables) on each arc indicating whether the arc is used or not. Each flow is bounded by a constant which corresponds to the production capacity.

$$X^{R1} = \{(x, y, s) \in \mathbb{R}_+^n \times \{0, 1\}^n \times \mathbb{R}_+^1 : \sum_{i \in N} x_i \leq b + s, x_i \leq P y_i, y_i \leq 1, 1 \leq i \leq n\}$$

The construction of one set X^{R1} from X^{LS-SW} is done by summing up constraint (1) over $\{t, \dots, k\}$, $1 \leq t \leq k \leq n$ and considering $s_{t-1} \geq 0$. Therefore, there exist n^2 relaxations of this type. Constraints (3) and (5) are eliminated. Production and vehicle capacities are also assumed to be constant. For each interval $[t, l]$ where $1 \leq t \leq l \leq n$, one can construct a relaxation of type X^{R1} from X^{LS-SW} , which makes n^2 relaxations in total.

Proposition 1 C_p is a minimal flow cover for X^{R1} if $\sum_{i \in C_p} P - b = \lambda^p$ with $\lambda^p > 0$. The lifted flow cover inequality

$$\sum_{i \in C_p} x_i + \sum_{i \in C_p} (P - \lambda^p)^+ (1 - y_i) + \sum_{i \in L} x_i - \sum_{i \in L} (P - \lambda^p)^+ y_i \leq b + s$$

where $L \subseteq N \setminus C_p$ is valid for X^{R1} .

Proof. Derived from MIR flow cover inequality in Louveaux and Wolsey [14] where $N_2 = \emptyset$. \square

In the case in which $s = 0$, Padberg *et al.* [15] have shown that an exponential number of flow cover facets suffice to characterize X^{R1} .

4.2.2 Integer flow cover for a valid relaxation of X^{LS-SW}

Mixed integer rounding approach can be used to derive MIR integer flow cover inequalities from the single node flow set. The first studies are from Atamtürk [2]. The author studies the mixed integer knapsack polyhedron which is defined by an arbitrary linear inequality and upper bounds on the integer and continuous variables. He describes facet defining inequalities obtained by a sequential lifting of inequalities with only one integer variable.

Here we give the second valid relaxation of X^{LS-SW} denoted by X^{R2} . The only difference between X^{R1} and X^{R2} is on the flow variables upper bound. In X^{R2} , flows are bounded by integer multiples of a constant and each integer variable is bounded by $\lceil \frac{P}{V} \rceil$. As an initial condition, we consider $\lceil \frac{P}{V} \rceil > 1$. Observe that for $P = V$, X^{R2} transforms into X^{R1} .

$$X^{R2} = \{(x, z, s) \in \mathbb{R}_+^n \times \mathbb{Z}_+^n \times \mathbb{R}_+ : \sum_{i \in N} x_i \leq b + s, x_i \leq V z_i, z_i \leq \lceil \frac{P}{V} \rceil, 1 \leq i \leq n\}$$

The construction of the set X^{R2} from the set X^{LS-SW} is done using the same transformation of constraint (1) (shown in the previous section), by the elimination of constraints (2), (4) and by the relaxation of constraint (5) (the binary variable is deleted to have the largest RHS). The production and the vehicle capacities are also assumed to be constant. The set X^{R2} is constructed from X^{LS-SW} for each interval $[t, \dots, l]$ where $1 \leq t \leq l \leq n$. Therefore, there are n^2 sets of type X^{R2} for LS-SW.

Proposition 2 C_v is an integer flow cover for X^{R2} if

i) $C_v \subset N$

ii) $\exists k \in C_v$ such that $\sum_{j \in C_v \setminus k} V \lceil \frac{P}{V} \rceil < b$ and \exists unique values λ^v and η such that $V\eta + \sum_{i \in C_v \setminus k} V \lceil \frac{P}{V} \rceil = b + \lambda^v$ with $0 < \lambda^v < V$, $\eta \in \mathbb{Z}^1$ and $1 \leq \eta \leq \lceil \frac{P}{V} \rceil$. Integer flow cover inequality

$$\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + \sum_{i \in C_v} (V - \lambda^v)(\lceil \frac{P}{V} \rceil - z_i) + \sum_{i \in L} x_i - \sum_{i \in L} (V - \lambda^v)z_i \leq b + s \quad (14)$$

where $L \subseteq N \setminus C_v$ is valid for X^{R2} .

Proof. Derived from the IFCI in Louveaux and Wolsey [14] assuming $N_2 = \emptyset$. \square

Remark 1 For b values with $b \bmod V = 0$, one can not find unique values λ^v and η respecting $0 < \lambda^v < V$, $\eta \in \mathbb{Z}^1$ and $1 \leq \eta \leq \lceil \frac{P}{V} \rceil$. An integer flow cover does not exist for this case.

Assuming $s = 0$, in a symmetric way as Padberg *et al.* [15], we show that X^{R2} plus integer flow cover inequalities give a complete linear inequality description of $\text{conv}(X^{R2})$.

Proposition 3 Let P^{R2} be the linear relaxation of X^{R2} plus the following set of integer flow cover inequalities for all subsets \mathfrak{S} with $|\mathfrak{S}| \geq |C_v|$,

$$IFCI(\mathfrak{S}) : \sum_{i \in \mathfrak{S}} x_i + (V - \lambda^v)(\lceil \frac{P}{V} \rceil |C_v| - \sum_{i \in \mathfrak{S}} z_i) \leq b + (V - \lambda^v)(\lceil \frac{P}{V} \rceil - \eta)$$

where $\mathfrak{S} \subseteq N$. P^{R2} completely describes $\text{conv}(X^{R2})$.

Proof. Let $M(\alpha, \beta)$ be the set of all optimal solutions for the problem $\text{OPT}(X^{R2}, (\alpha, \beta))$ where the objective function $\sum_{i \in N} (\alpha_i x_i - \beta_i z_i)$ is maximized over the set X^{R2} . Our aim is to prove that $P^{R2} = \text{conv}(X^{R2})$. First we show that $\dim(\text{conv}(X^{R2})) = \dim(P^{R2})$. Then, we demonstrate that for all values of α_i and β_i for which $M(\alpha, \beta) \neq X^{R2}$, the optimal value is finite and $M(\alpha, \beta) \subseteq \{(x, z) : ax - bz = c\}$ for some inequality $ax - bz \leq c$ defining P^{R2} such that $P^{R2} \cap \{(x, z) : ax - bz = c\}$ is a proper face of P^{R2} . The condition $M(\alpha, \beta) \neq X^{R2}$ implies that the cost vectors α and β can not both have all their components null, since it is trivial that $M(0, 0) = X^{R2}$.

For positive values of b and $\lceil \frac{P}{V} \rceil$, for each variable, there exists a non-empty real interval in which the latter can take any value independently from the others in $\text{conv}(X^{R2})$. Observe that $\text{conv}(X^{R2}) \subseteq P^{R2} \subseteq \mathbb{R}^{2n}$. Therefore, $\dim(\text{conv}(X^{R2})) = \dim(P^{R2}) = 2n$. We show below that for any value of (α, β) , the optimal solution satisfies one of the inequality in P^{R2} as equality and the intersection of P^{R2} with each of these equalities defines a proper face for P^{R2} . The objective function to optimize is thus,

$$\max(\sum_{i \in N} (\alpha_i x_i - \beta_i z_i))$$

Below we denote by α_i and β_i the i^{th} component of the cost vector (α, β) .

- If $\exists i$ with $\beta_i < 0$, then $M(\alpha, \beta) \subseteq \{(x, z) : z_i = \lceil \frac{P}{V} \rceil\}$
- If $\exists i$ with $\alpha_i < 0$, then $M(\alpha, \beta) \subseteq \{(x, z) : x_i = 0\}$
- If $\exists i$ with $\alpha_i = 0$ and $\beta_i > 0$, then $M(\alpha, \beta) \subseteq \{(x, z) : z_i = 0\}$. More generally, if $V\alpha_i < \beta_i$, then $M(\alpha, \beta) \subseteq \{(x, z) : z_i = 0\}$.

By now, we assume that none of these cases happens. Therefore, we are in the situation in which $\alpha_i \geq 0, \beta_i \geq 0$ and $V\alpha_i \geq \beta_i, \forall i \in N$. We partition N into \mathfrak{S} and \mathfrak{S}' such that $\mathfrak{S} = \{i : \alpha_i > 0\}$ and $\mathfrak{S}' = \{i : \alpha_i = 0\}$. Notice that $\forall i \in \mathfrak{S}'$ we have $\beta_i = 0$. This implies that $\mathfrak{S} \neq \emptyset$, since there must exist at least one component with $\alpha_i \neq 0$, otherwise we do not satisfy the initial condition $M(\alpha, \beta) \neq X^{R2}$. In addition, $\exists i \in \mathfrak{S}$ such that $V\alpha_i = \beta_i$, then $M(\alpha, \beta) \subseteq \{(x, z) : x_i = Vz_i\}$. Hence we have $V\alpha_i > \beta_i$ on \mathfrak{S} . We consider two cases depending on whether \mathfrak{S} is a cover or not.

- If $|\mathfrak{S}| \lceil \frac{P}{V} \rceil \leq b$ then $M(\alpha, \beta) \subseteq \{(x, z) : x_i = Vz_i\}$
- If $|\mathfrak{S}| \lceil \frac{P}{V} \rceil > b$ then
 - If $\exists i \in \mathfrak{S}$ with $\beta_i = 0$ then $M(\alpha, \beta) \subseteq \{(x, z) : \sum_{i \in \mathfrak{S}} x_i = b\}$
 - Otherwise any optimal solution verifies $\sum_{i \in \mathfrak{S}} x_i = b$ or $\sum_{i \in \mathfrak{S}} x_i = \lfloor \frac{b}{V} \rfloor V$ and $\sum_{i \in \mathfrak{S}} z_i = (|C_v| - 1) \lceil \frac{P}{V} \rceil + \eta$ or $\sum_{i \in \mathfrak{S}} z_i = \lfloor \frac{b}{V} \rfloor$. It implies that IFCI(\mathfrak{S}) is tight. $M(\alpha, \beta) \subseteq \{(x, z) : \sum_{i \in \mathfrak{S}} x_i + (V - \lambda^v)(\lceil \frac{P}{V} \rceil |C_v| - \sum_{i \in \mathfrak{S}} z_i) = b + (V - \lambda^v)(\lceil \frac{P}{V} \rceil - \eta)\}$.

It can easily be checked that all the faces used in the proof are proper faces of P^{R2} . \square

4.3 New valid inequalities: Mixed flow cover

The third valid relaxation of X^{LS-SW} corresponds to $(X^{R1} \cap X^{R2})$ with the addition of lower and upper bounds on the variables z_i (see constraints (4) and (5) in X^{LS-SW} formulation). We denote by X^{mix} this new set. Since sets X^{R1} and X^{R2} are valid relaxations for the initial set X^{LS-SW} , their intersection is also valid. Therefore, there are n^2 valid relaxations of this type (one set for each interval $\{t, \dots, l\}$, $1 \leq t \leq l \leq n$). In this section, we derive non-trivial facets for X^{mix} by lifting IFCI. Remember that the IFCI can be derived for the case in which $\lceil \frac{P}{V} \rceil > 1$. Thus, to obtain the mixed flow cover inequality (MFCI) we also make this assumption.

Definition 1 X^{mix} is the set of points $(x, y, z, s) \in \mathbb{R}_+^n \times \{0, 1\}^n \times \mathbb{Z}_+^n \times \mathbb{R}_+$ satisfying

$$\begin{aligned} \sum_{i \in N} x_i &\leq b + s \\ x_i &\leq Py_i, \quad i \in N \\ x_i &\leq Vz_i, \quad i \in N \\ y_i &\leq z_i \leq \lceil \frac{P}{V} \rceil y_i, \quad i \in N \end{aligned}$$

We first establish the dimension and exhibit trivial facets of X^{mix} , then we show the validity of the mixed flow cover inequalities for X^{mix} for the case $P \bmod V = 0$.

Proposition 4 X^{mix} is full dimensional: $\dim(X^{mix}) = 3n + 1$.

Proof. The number of variables in X^{mix} is $3n + 1$. Therefore, $\dim(X^{mix}) \leq 3n + 1$. It suffices to find $3n + 2$ affinely independent points in X^{mix} . We assume $b > 0$. See Algorithm 1 to generate $3n$ points. The other two feasible points are; $x_j = 0, y_j = 0, z_j = 0, s = 0, \forall j \in \{1, \dots, n\}$ and $x_j = 0, y_j = 0, z_j = 0, s = 1, \forall j \in \{1, \dots, n\}$. Thus, $\dim(X^{mix}) \geq 3n + 1$, which gives the dimension of X^{mix} , $\dim(X^{mix}) = 3n + 1$. \square

Proposition 5 Trivial facets of X^{mix} are

1. $x_i \geq 0, \forall i \in N$
2. $y_i \leq z_i, \forall i \in N$

Algorithm 1 Generation of $3n$ feasible affinely independent points for X^{mix} .

```

 $\delta = \begin{cases} (b \bmod P) & \text{if } (b \bmod P) > 0, \\ P & \text{else.} \end{cases}$ 
for all  $k$  such that  $1 \leq k \leq n$  do
  for all  $j$  such that  $1 \leq j \leq n$  do
    if  $j = k$  then
       $x_j \leftarrow \delta, y_j \leftarrow 1, z_j \leftarrow \lceil \frac{\delta}{V} \rceil, s \leftarrow 0$ 
    else
       $x_j \leftarrow 0, y_j \leftarrow 0, z_j \leftarrow 0, s \leftarrow 0$  or
       $x_j \leftarrow 0, y_j \leftarrow 1, z_j \leftarrow 1, s \leftarrow 0$  or
       $x_j \leftarrow 0, y_j \leftarrow 1, z_j \leftarrow \lceil \frac{P}{V} \rceil, s \leftarrow 0$ 
    end if
  end for
end for

```

$$3. \ z_i \leq \lceil \frac{P}{V} \rceil y_i, \forall i \in N$$

$$4. \ x_i \leq Vz_i, \forall i \in N, \text{ for } b \geq 2V$$

$$5. \ y_i \leq 1, \forall i \in N$$

$$6. \ \sum_{i \in N} x_i \leq b, \text{ for } |C_p| < n$$

Proof. For each of these facets, $3n + 1$ affinely independent points are given in Appendix A. For $b < 2V$, $x_i \leq by_i$ becomes facet defining and replaces the 4th inequality which is redundant. \square

In proposition 6, we prove the validity of the mixed flow cover inequality. We assume $P \bmod V = 0$ to simplify the computation of the lifting coefficients. This result is extended in proposition 10 for the case $P \bmod V \neq 0$ and in proposition 13 for the case in which P_i and V_i are time varying. Remark that for $P \bmod V = 0$, we have $|C_v| = |C_p|$ ($|C_p|$ is the number of variables in the flow cover), which is not the case for $P \bmod V \neq 0$. This equality can be shown as follows. Observe that $(|C_p| - 1)P + P - \lambda^p = b$ for the flow cover inequality and $(|C_v| - 1)P + V\eta - \lambda^v = b$ for the integer flow cover inequality. $P - \lambda^p$ and $V\eta - \lambda^v$ are the quantities which exceed $\lfloor \frac{b}{P} \rfloor$ in the flow and integer flow cover inequalities respectively. These two quantities are then equal, therefore we have $|C_v| = |C_p|$.

Proposition 6 *If C_v is an integer flow cover ($b \bmod V \neq 0$ is assumed) then, the mixed flow cover inequality (MFCI)*

$$\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) + \lambda^v(\eta - 1) \sum_{i \in C_v} (1 - y_i) \leq b + s \quad (15)$$

is valid for X^{mix} for $P \bmod V = 0$.

Proof. The validity of the IFCI (14) for X^{R2} implies also its validity for X^{mix} which is more restricted than X^{R2} . The MFCI (15) is derived from the IFCI assuming $L = \emptyset$, by a sequence independent lifting procedure. The associated face to inequality (14) can be lifted to a higher dimension adding the slack variables $(1 - y_i)$ with a certain coefficient α_i in the left hand side.

Let us first add $\alpha_1(1 - y_1)$. For $y_1 = 1$, there is no change. For $0 \leq y_1 < 1$, a positive amount will be added in the left hand side.

$$\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) + \alpha_1(1 - y_1) \leq b + s$$

The aim is to find the maximum value of α_1 which makes the inequality the strongest possible, without violating the initial constraints of the set X^{mix} . We can eliminate the variable s during the computation of the lifting coefficients, since the exceeding amount of x variables goes into s variable, which makes their difference 0. Hence,

$$\alpha_1 = b - \max\{\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) : y_1 = 0, (x, y, z) \in X^{mix}\}$$

Notice that $y_1 = 0 \Rightarrow x_1 = 0, z_1 = 0$. By definition of the flow cover, we have $\sum_{i \in C_p \setminus 1} P < b$. The constraint $\sum_{i \in N} x_i \leq b$ is thus redundant. The latter allows us to saturate all the other x variables up to P , to realize the maximal value.

$$\alpha_1 = b - [(|C_v| - 1)P + (V - \lambda^v)(\eta - \frac{P}{V}) + (V - \lambda^v)(\frac{P}{V})]$$

Using the definition of the integer flow cover, one can write $V\eta + (|C_v| - 1)P = b + \lambda^v$, hence,

$$\alpha_1 = -(\lambda^v + P - V\eta) + P - (V - \lambda^v)\eta = \lambda^v(\eta - 1)$$

For the second variable, we proceed similarly but this time paying attention to the new term added. In order to obtain the maximal value, again, all x values are saturated to P except x_2 . Consider that x variables are not saturated and they take values less than P . If $0 \leq x_i < P$ then it is trivial that the maximal value obtained decreases.

$$\alpha_2 = b - \max\{\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) + \lambda^v(\eta - 1)(1 - y_1) : y_2 = 0, (x, y, z) \in X^{mix}\} = \lambda^v(\eta - 1)$$

For any lifting sequence, the same coefficients are found, thus the lifting is sequence independent. More formally, considering the following valid inequality, let us compute the value of α_{j+1} .

$$\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) + \lambda^v(\eta - 1) \sum_{i=1}^j (1 - y_i) \leq b + s$$

$$\alpha_{j+1} = b - \max\{\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + \sum_{i \in C_v} (V - \lambda^v)(\frac{P}{V} - z_i) + \lambda^v(\eta - 1) \sum_{i=1}^j (1 - y_i) : y_{j+1} = 0, (x, y, z) \in X^{mix}\} = \lambda^v(\eta - 1)$$

Therefore, the new class of valid inequalities, that we call *mixed flow cover inequalities (MFCI)*, can be written in the form (15) for $P \bmod V = 0$. \square

Observation 1 *Inequality (15) is valid for X^{mix} because of the construction procedure of the lifting coefficients. They are all computed satisfying the initial constraints of X^{mix} .*

Proposition 7 *(15) is facet defining for the set X^{mix} if $b \bmod V \neq 0$ and $P \bmod V = 0$.*

Proof. The validity of the inequality is known. To show that it is facet defining, $3n + 1$ affinely independent points satisfying inequality (15) as equality have to be found. There are two cases: $n = |C_v|$ and $n > |C_v|$. For each case, $3n$ linearly independent directions are shown in Appendix B1, respectively in Tables 13 and 14. Each direction is the difference of two feasible solutions for X^{mix} which satisfy inequality (15) as equality.

For the first case where $n = |C_v|$, in order to find $2|C_v|$ affinely independent points, the idea is to saturate all x variables to P , except one which takes either the value $P - \lambda^p$ or 0. For the last $|C_v|$ points, if $\eta > 1$, all x variables are saturated to P , except one which is set to $(\eta - 1)V$. Otherwise, if $\eta = 1$, all the variables except two are saturated to P , and the two variables take values $P - V$ and V . To obtain the last point, for one of the previous points, the value of s variable is increased by one unit.

For the second case where $n > |C_v|$, $3|C_v| + 1$ directions are obtained as previously. In order to obtain $3(n - |C_v|)$ other directions we proceed as follows. To obtain $n - |C_v|$ directions, one can substitute the value of each non-cover element in $N \setminus C_v$ whose value is 0 by the cover element in C_v whose value is $P - \lambda^p$. To obtain another $n - |C_v|$ directions, y variables for the non-cover elements can take the value 1 instead of 0. For the last $n - |C_v|$ directions, the value of the z variables in $N \setminus C_v$ are increased to their maximum value $\lceil \frac{P}{V} \rceil > 1$. As the terms in $N \setminus C_v$ do not appear in the inequality, the change on the values of the non-cover variables does not influence the LHS value. \square

Proposition 8 *The MFCI (15) can be lifted using variables in $L \subseteq N \setminus C_v$. The lifted MFCI*

$$\sum_{i \in C_v} x_i + (V - \lambda^v)(\eta - \frac{P}{V}) + (V - \lambda^v) \sum_{i \in C_v} (\frac{P}{V} - z_i) + \sum_{i \in C_v} \lambda^v(\eta - 1)(1 - y_i) + \sum_{i \in L} x_i - (V - \lambda^v) \sum_{i \in L} z_i - \sum_{i \in L} \lambda^v(\eta - 1)y_i \leq b + s \quad (16)$$

where $L \subseteq N \setminus C_v$ is valid for X^{mix} .

Proof. Let us rewrite the non-lifted inequality with all variable components in the LHS and the constant parts in the RHS.

$$\sum_{i \in C_v} (x_i - (V - \lambda^v)z_i - \lambda^v(\eta - 1)y_i) - s \leq b - [(V - \lambda^v)(\eta - \frac{P}{V}) + (V - \lambda^v)\frac{P}{V}|C_v| + \lambda^v(\eta - 1)|C_v|] \quad (17)$$

Without loss of generality, we may assume that $s = 0$. Observe that the maximal value of the LHS can be obtained for different values of the variables. See the proof of proposition 9 for the points which make (17) tight. The RHS value can be attained using only the variables in C_v . Thus, $|C_v|$ variables are enough to saturate $\sum_{i=1}^N x_i = b$. Now, let us write the lifted form as,

$$\sum_{i \in C_v \cup L} (x_i - (V - \lambda^v)z_i - \lambda^v(\eta - 1)y_i) - s \leq b - [(V - \lambda^v)(\eta - \frac{P}{V}) + (V - \lambda^v)\frac{P}{V}|C_v| + \lambda^v(\eta - 1)|C_v|] \quad (18)$$

Compared to inequality (17), we add the term $\sum_{i \in L} (x_i - (V - \lambda^v)z_i - \lambda^v(\eta - 1)y_i)$ in the LHS and the RHS remains unchanged. Now consider that one opens a production period j in L and decides to produce less in some periods $j' \in C_v$ in order to satisfy $\sum_{i=1}^N x_i = b$. If one continues to produce in each period in C_v , then we have at least an additional setup period and maybe an additional batch. The new LHS in (18) can only be less than or equal to the previous one in (17), because y and z variables have negative coefficients. Therefore, to maximize the LHS value, one period in C_v must become free. Observe that a substitution is done between the new production period j in L and the old production period j' in C_v , which shows the validity of (18). \square

Proposition 9 *The lifted MFCI (15) is facet defining for the set X^{mix} in most cases if $b \bmod V \neq 0$, $P \bmod V = 0$ and $|C_v| < N$.*

Proof. Respectively $3|C_v| + 1$ and $3(n - |C_v|)$ directions are given in Appendix B1 in Tables 13 and 14 for the case $L = \emptyset$. For the case $L \neq \emptyset$, the same $3|C_v| + 1$ and $3(n - |C_v| - L)$ directions can be used. Additionally, we substitute the value of the variables in the cover with the 0 valued variables in L . The value of the x variables to interchange are respectively $P - \lambda^p$, $V(\eta - 1)$ and P if $\eta > 1$, V otherwise. The new matrix obtained with the positive non-cover elements in L has exactly the same stages as the one obtained with the cover elements in C_v . \square

We showed the validity of the MCFIs with and without lifting for the case $P \bmod V = 0$ and identified the cases where they are facet defining for X^{mix} . Now, we remove the restriction $P \bmod V = 0$, and show the validity of the MFCIs. P and V are again assumed to be constant.

Proposition 10 *If C_v is an integer flow cover, then for the case $P \bmod V \neq 0$, the MFCI*

$$\sum_{i \in C_v \cup S} x_i + (V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + \sum_{i \in C_v} (V - \lambda^v)(\lceil \frac{P}{V} \rceil - z_i) - \sum_{i \in S} (V - \lambda^v)z_i + \beta(1 - y_1) + \sum_{i \in (C_v \cup S) \setminus \{1\}} \alpha(1 - y_i) \leq b + s \quad (19)$$

where $1 \in C_v$, $C_v \subseteq C_p$, $S \subseteq C_p \setminus C_v$, $\beta = b_v - \phi^\Sigma + \phi$, $\alpha = \min\{\beta, \phi\}$, $b_v = b - ((V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + (V - \lambda^v)\lceil \frac{P}{V} \rceil|C_v|)$, $\phi = \max\{P - (V - \lambda^v)\lceil \frac{P}{V} \rceil; \lambda^v \lfloor \frac{P}{V} \rfloor\}$, $\phi^\Sigma = \sum_{C_v \cup S} \phi$, is valid for X^{mix} .

Proof. (We order the index of the cover variables in C_v according to the sequence of lifting). For $P \bmod V \neq 0$, the computation of the lifting coefficients becomes sequence dependent. The first slack variable $(1 - y_1)$ to lift has always a coefficient β greater than or equal to the coefficients α_i of the variables which follow in the lifting computation. Another important change is on the number of integer flow cover elements $|C_v|$. For the case $P \bmod V = 0$, we have shown that $|C_v| = |C_p|$ but for $P \bmod V \neq 0$, we only have $|C_v| \leq |C_p|$. For $P \bmod V \neq 0$, we observed in PORTA² that most of the mixed flow cover type inequalities defining a facet for X^{mix} are obtained by lifting an initial IFCI containing already all the variables in C_p . It is also possible to lift the IFCI containing only the variables in $C_v \cup S'$ with $S' \subset C_p \setminus C_v$ to obtain a valid inequality. The compromise is then either to obtain an inequality with less variables $|C_v \cup S|$ but a bigger β or with more variables $|C_p|$ but a smaller β . Here we assume the more general case where $S \subseteq C_p \setminus C_v$.

Consider the initial IFCI in which we add the first lifted term with a coefficient β .

$$\sum_{i \in C_v \cup S} x_i + (V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + \sum_{i \in C_v} (V - \lambda^v)\lceil \frac{P}{V} \rceil - \sum_{i \in C_v \cup S} (V - \lambda^v)z_i + \beta(1 - y_1) \leq b + s$$

Let us compute the biggest β , for which the inequality satisfies the initial constraints in X^{mix} and remains valid. We denote by b_v the following constant,

$$b_v = b - ((V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + (V - \lambda^v)\lceil \frac{P}{V} \rceil|C_v|)$$

Therefore, one can write

$$\beta = b_v - \max\left\{\sum_{i \in C_v \cup S} (x_i - (V - \lambda^v)z_i) : y_1 = 0, (x, y, z) \in X^{mix}\right\}$$

When $y_1 = 0$, the constraint $\sum_{i \in N} x_i \leq b$ becomes redundant, since $\sum_{i \in (C_v \cup S) \setminus \{1\}} P \leq \sum_{i \in (C_p) \setminus \{1\}} P < b$ by definition. Therefore, we can compute the maximal value of each term $x_i - (V - \lambda^v)z_i$ independently for all i . Let $\phi^i = \max\{x_i - (V - \lambda^v)z_i : (x, y, z) \in X^{mix}\}$ and $\phi^\Sigma = \sum_{C_v \cup S} \phi^i$. Hence

$$\beta = b_v - \phi^\Sigma + \phi^1$$

²Polyhedron Representation Transformation Algorithm (by Thomas Christof, Heidelberg Univ.). Routine collection to analyze polytopes and polyhedra. <http://www.zib.de/Optimization/Software/Porta>.

We describe the computation ϕ^i at the end of the proof. Now, let us compute α_2 .

$$\alpha_2 = b_v - \max\left\{ \sum_{i \in C_v \cup S} (x_i - (V - \lambda^v)z_i) + \beta(1 - y_1) : y_2 = 0, (x, y, z) \in X^{mix} \right\}$$

There are $2^1 = 2$ possibilities, either $y_1 = 0$ or $y_1 = 1$.

- If $y_1 = 0$ then

$$\begin{aligned} \alpha_2^{(1)} &= b_v - (\max_{C_v \cup S} \{x_i - (V - \lambda^v)z_i : y_1 = 0, y_2 = 0, (x, y, z) \in X^{mix}\} + \beta) \\ \alpha_2^{(1)} &= b_v - \phi^\Sigma + \phi^1 + \phi^2 - \beta = \phi^2 \end{aligned}$$

- If $y_1 = 1$ then

$$\begin{aligned} \alpha_2^{(2)} &= b_v - (\max_{C_v \cup S} \{x_i - (V - \lambda^v)z_i + \beta(1 - y_1) : y_1 = 1, y_2 = 0, (x, y, z) \in X^{mix}\}) \\ \alpha_2^{(2)} &= b_v - \phi^\Sigma + \phi^2 \end{aligned}$$

$$\alpha_2 = \min\{\alpha_2^{(1)}, \alpha_2^{(2)}\} = \min\{\phi^2, b_v - \phi^\Sigma + \phi^2\}$$

Now, let us compute α_3 ,

$$\alpha_3 = b_v - \max\left\{ \sum_{i \in C_v \cup S} (x_i - (V - \lambda^v)z_i) + \beta(1 - y_1) + \alpha_2(1 - y_2) : y_3 = 0, (x, y, z) \in X^{mix} \right\}$$

There are $2^2 = 4$ possibilities, either $(y_1 = 1, y_2 = 1)$ or $(y_1 = 0, y_2 = 1)$ or $(y_1 = 1, y_2 = 0)$ or $(y_1 = 0, y_2 = 0)$. We compute α_3 for each case.

- If $y_1 = 1, y_2 = 1$ then $\alpha_3^{(1)} = b_v - \phi^\Sigma + \phi^3$
- If $y_1 = 0, y_2 = 1$ then

$$\begin{aligned} \alpha_3^{(2)} &= b_v - (\max_{C_v \cup S} \{x_i - (V - \lambda^v)z_i : y_1 = y_3 = 0, y_2 = 1, (x, y, z) \in X^{mix}\}) - \beta \\ \alpha_3^{(2)} &= b_v - \phi^\Sigma + \phi^1 + \phi^3 - \beta = \phi^3 \end{aligned}$$

- If $y_1 = 1, y_2 = 0$ then

$$\begin{aligned} \alpha_3^{(3)} &= b_v - (\max_{C_v \cup S} \{x_i - (V - \lambda^v)z_i : y_2 = y_3 = 0, y_1 = 1, (x, y, z) \in X^{mix}\}) - \alpha_2 \\ \alpha_3^{(3)} &= b_v - \phi^\Sigma + \phi^2 + \phi^3 - \alpha_2, \text{ as } \alpha_2 = \min\{\phi^2, b_v - \phi^\Sigma + \phi^2\} \Rightarrow \phi^2 \geq \alpha_2, \text{ then} \\ \alpha_3^{(3)} &\geq b_v - \phi^\Sigma + \phi^3 = \alpha_3^{(1)} \end{aligned}$$

- If $y_1 = 0, y_2 = 0$ then

$$\begin{aligned} \alpha_3^{(4)} &= b_v - (\max_{C_v \cup S} \{x_i - (V - \lambda^v)z_i + \sum_{i=1}^2 \alpha_i(1 - y_i) : y_1, y_2, y_3 = 0, (x, y, z) \in X^{mix}\}) \\ \alpha_3^{(4)} &= b_v - \phi^\Sigma + \phi^1 + \phi^2 + \phi^3 - \beta - \alpha_2 \Rightarrow \alpha_3^{(4)} \geq \alpha_3^{(2)} \end{aligned}$$

Therefore, $\alpha_3 = \min\{\alpha_3^{(1)}, \alpha_3^{(2)}, \alpha_3^{(3)}, \alpha_3^{(4)}\} = \min\{\phi^3, b_v - \phi^\Sigma + \phi^3\}$.

More generally, we consider $\alpha_j = \min\{\phi^j, b_v - \phi^\Sigma + \phi^j\}$ and compute α_{j+1} .

$$\alpha_{j+1} = b_v - \max\left\{\sum_{i \in C_v \cup S} (x_i - (V - \lambda^v)z_i) + \sum_{i=1}^j \alpha_i(1 - y_i) : y_{j+1} = 0, (x, y, z) \in X^{mix}\right\}$$

There are 2^j possibilities to set the variables $y_i, \forall i \in \{1, \dots, j\}$, y_{j+1} is already fixed to 0. In the computation of α_{j+1} , for each variable i such that $y_i = 0$, the term $\phi_i - \alpha_i$ is added, otherwise there is no addition. Since $\alpha_i = \min\{\phi^i, b_v - \phi^\Sigma + \phi^i\}$, then we have $\phi_i - \alpha_i \geq 0$. For each variable $i \in C_v \cup S$ except the first one, α_{j+1} becomes bigger with the addition of this term. The aim being to find the minimum value of α_{j+1} , these cases can be eliminated. Thus, two vectors are important in the comparison: $(1, 1, \dots, 1)$ and $(0, 1, 1, \dots, 1)$. One can conclude that $\alpha_{j+1} = \min\{\phi^{j+1}, b_v - \phi^\Sigma + \phi^{j+1}\}$.

Let us now compute the values $\phi_i = \max\{x_i - (V - \lambda^v)z_i : (x, y, z) \in X^{mix}\}, \forall i \in N$ independently. One can easily remark that ϕ^i has its maximal value when $x_i = P$ (saturation of the production capacity) or $x_i = \lfloor \frac{P}{V} \rfloor V$ (saturation of the maximum number of vehicles). Hence

$$\phi^i = \max\{P - (V - \lambda^v)\lceil \frac{P}{V} \rceil; \lambda^v \lfloor \frac{P}{V} \rfloor\}$$

For P and V constant, $\phi^i = \phi = \max\{P - (V - \lambda^v)\lceil \frac{P}{V} \rceil; \lambda^v \lfloor \frac{P}{V} \rfloor\}$, for all $i \in N$. This allows to write $\beta = b_v - \phi^\Sigma + \phi$ and $\alpha = \min\{\beta, \phi\}$. This procedure yields a valid inequality, since in each step all the initial constraints of X^{mix} are satisfied. \square

Proposition 11 *If C_v is an integer flow cover, $L \subseteq N \setminus (C_v \cup S)$ and $\beta \geq \phi$, the lifted MFCI*

$$\begin{aligned} \sum_{i \in C_v \cup S \cup L} x_i + (V - \lambda^v)(\eta - \lceil \frac{P}{V} \rceil) + \sum_{i \in C_v} (V - \lambda^v)(\lceil \frac{P}{V} \rceil - z_i) - \sum_{i \in S \cup L} (V - \lambda^v)z_i + \beta(1 - y_1) \\ + \sum_{i \in (C_v \cup S) \setminus \{1\}} \alpha(1 - y_i) - \sum_{i \in L} \alpha y_i \leq b + s \end{aligned} \quad (20)$$

is valid for X^{mix} .

Proof. Let us rewrite the non-lifted integer flow cover inequality (19) by putting all the variables in the LHS and all the constant components in the RHS.

$$\sum_{i \in C_v \cup S} x_i - \sum_{i \in C_v \cup S} (V - \lambda^v)z_i - \beta y_1 - \sum_{i \in (C_v \cup S) \setminus \{1\}} \alpha y_i - s \leq b_v - \beta - (|C_v \cup S| - 1)\alpha$$

Compared to the above inequality, the lifted MFCI contains the term $\Omega = \sum_{i \in L} (x_i - (V - \lambda^v)z_i - \alpha y_i)$ in its LHS. If $\beta \geq \phi$, then $\alpha = \phi$. Therefore, $\max_{i \in L} (x_i - (V - \lambda^v)z_i - \alpha y_i) \leq 0$ by definition of ϕ , which means that the LHS value can not be increased by the addition of Ω , which shows the validity of (20). Observe that, in this case the lifting is useless due to the negative or null terms added in the LHS. \square

Observation 2 *The lifted MFCI (20) reduces to the IFCI (14) for $\eta = 1$.*

Example 1 On a simple example, we show how the MFCI cuts a fractional point which can not be cut by the FCI or IFCI. Let us denote by X^{e1} the following set,

$$\begin{aligned} & \max(x_1 + x_2 + x_3 - 2y_1 - 2y_2 - 2y_3 - 2z_1 - 2z_2 - 2z_3) \\ & \sum_{i=1}^3 x_i \leq 26 \\ & x_i \leq 9y_i \\ & x_i \leq 3z_i \\ & y_i \leq z_i \leq 3y_i \\ & (x, y, z) \in \mathbb{R}_+^3 \times \{0, 1\}^3 \times \mathbb{Z}_+^3 \end{aligned}$$

See in Table 1 for the solutions of the linear relaxation of X^{e1} , with or without the addition of the cover inequalities. The objective function value obtained after the addition of the MFCI is better than the others.

cases	x_1	x_2	x_3	y_1	y_2	y_3	z_1	z_2	z_3	OF
no ineq. added	8	9	9	$\frac{8}{9}$	1	1	$\frac{8}{9}$	3	3	2.88
only flow cover	8	9	9	1	1	1	$\frac{8}{9}$	3	3	2.66
only integer flow cover	9	6	9	1	$\frac{6}{9}$	1	3	2	3	2.66
flow+int. flow cov.	9	6	9	1	0.75	1	3	2	3	2.5
only mix. flow cov.	9	9	8	1	1	1	3	3	3	2*

Table 1: Linear relaxation values for different cases.

For large values of n , the number of cover inequalities to add in the initial formulation of LS-SW can be quite high. In order to add these inequalities only when they are needed, we propose a cutting plane algorithm described in the next section. Before introducing the separation algorithm, some statistics are given to have an idea on the number of different facet defining inequalities obtained by PORTA for the set X^{mix} . For the test results, the instance of Example 1 is used. The complete list for all b values can be found in Appendix C1, in Table 16. For $b = 14$, we provide also all facet defining inequalities in Appendix D.

The results in Table 16 are obtained for X^{e1} when $P \bmod V = 0$. Different tests are also carried out for the case $P \bmod V \neq 0$. We found out that for this case, there are many more inequalities which are not of cover type. We give Table 17 in Appendix C2 for $P = 9$, $V = 4$. In this case, a new facet defining inequalities appears. Other trivial facets for $P \bmod V \neq 0$ remain the same as in proposition 5.

Proposition 12 For $P \bmod V \neq 0$, the inequality

$$x_i \leq \varepsilon y_i + \mu z_i, \text{ where } \begin{cases} \mu = P \bmod V \\ \varepsilon = P - \lceil \frac{P}{V} \rceil \mu \end{cases} \text{ defines a facet for } X^{mix}$$

Proof. $3n + 1$ affinely independent points satisfying this inequality as equality are given in Appendix A. \square

Before describing the cutting plane algorithm, we state a final proposition for the case where the production and the vehicle capacities take time varying values.

Proposition 13 If C_v is an integer flow cover, for the case in which P_i , V_i take time varying

values and if $\sum_{i \in C_v \setminus j} P_i < b, \forall j \in C_v$, then, the MFCI

$$\begin{aligned} \sum_{i \in C_v \cup S} x_i + (V_k - \lambda^v)^+ (\eta_k - \lceil \frac{P_k}{V_k} \rceil) + \sum_{i \in C_v} (V_i - \lambda^v)^+ (\lceil \frac{P_i}{V_i} \rceil - z_i) - \sum_{i \in S} (V_i - \lambda^v)^+ z_i + \beta(1 - y_1) \\ + \sum_{i \in (C_v \cup S) \setminus \{1\}} \alpha_i(1 - y_j) \leq b + s \end{aligned} \quad (21)$$

where $1 \in C_v$, $C_v \subseteq C_p$, $S \subseteq C_p \setminus C_v$, $\beta = b_v - \phi^\Sigma + \phi^1$, $\alpha_i = \min\{b_v - \phi^\Sigma + \phi^i, \phi^i\}$, $b_v = b - ((V_k - \lambda^v)^+ (\eta_k - \lceil \frac{P_k}{V_k} \rceil) + \sum_{i \in C_v} (V_i - \lambda^v)^+ \lceil \frac{P_i}{V_i} \rceil)$, $\phi^i = \max\{P_i - (V_i - \lambda^v)^+ \lceil \frac{P_i}{V_i} \rceil; \lambda^v \lfloor \frac{P_i}{V_i} \rfloor\}$, $\phi^\Sigma = \sum_{i \in C_v \cup S} \phi^i$, is valid for X^{mix} .

Proof. One can proceed in the same manner as in proposition 10. Each time we compute the coefficient α_j for a slack variable $(1 - y_j)$, we give the 0 value to y_j . The fact that $\forall j \in C_v$, $\sum_{i \in C_v \setminus j} P_i < b$ allows us to saturate the other variables values in order to compute α_j . The first change compared to proposition 10 is the construction of the initial integer flow cover set. This one depends on the choice of the variable k . For different k values, C_v, λ^v and η_k will be different. Once an integer flow cover inequality is found, the computation of β and α_i remains thus the same. \square

Remark 2 If the initial assumption $\forall j \in C_v, \sum_{i \in C_v \setminus j} P_i < b$ does not hold, then we have to solve an NP-hard optimization problem in order to compute the lifting coefficients. The constraints of the problem are exactly the same as in X^{mix} . This optimization problem takes the form:

$$\begin{aligned} \max \quad & \sum_{i \in C_v \setminus j} (x_i - (V_i - \lambda^v)^+ z_i - \alpha_i y_i) \\ \sum_{i \in C_v \setminus j} \quad & x_i \leq b + s \\ & \text{Other constraints of } X^{mix} \end{aligned}$$

In this paper, we do not give further details on this second case.

5 Cutting plane algorithm

The separation problem for cover inequalities

Given (x^*, y^*, z^*, s^*) satisfying X^{LS-SW} , either find a cover inequality (FCI, IFCI or MFCI) cutting off the fractional point, or show that all inequalities are satisfied. In LS-SW, one can find n^2 valid relaxations of type X^{mix} (see Definition 1). Therefore, for each interval $[k, l]$ with $1 \leq k \leq l \leq n$, a separation problem is solved for $X_{[k, l]}^{mix}$. For the set $X_{[k, l]}^{mix}$, $N = \{k, \dots, l\}$. Algorithm 2 and 3 constitute subprocedures for Algorithm 4. Algorithm 2 provides the computation of all the parameters to be used for generating the cover inequalities for each interval $[k, l]$. Algorithm 3 is written to compute the constant and variable components in the LHS of each cover inequality. In Algorithms below P and V are assumed to be constant. At the end of this section, the idea of the basic differences and similarities will be given for the case in which P_i and V_i can take time varying values.

Once all the parameters are computed in Algorithm 2 for each interval $\{t, \dots, l\}$, then the cover sets must be chosen. The aim of the separation algorithm is to find the most violated inequalities. Therefore, one has to choose the cover variables with the biggest variable component values. $f(i)$, $I(i)$ and $m(i)$ (see Algorithm 3) are sorted in a decreasing order in $O(n \log(n))$ time, and their indices are saved in tables tf , tI and tm .

Algorithm 2 Computation of the number of the cover elements, flow and integer flow cover excess and η for $X_{[k,l]}^{mix}$

Require: t, l, d, P, V

Ensure: $n, b, C_p, C_v, \lambda^p, \lambda^v, \eta, \alpha, \beta$

```

for all  $k$  such that  $1 \leq k \leq n$  do
  for all  $l$  such that  $k \leq l \leq n$  do
     $b = \sum_{i \in \{k, \dots, l\}} d_i$ 
     $|C_p| = \lceil \frac{b}{P} \rceil$  {flow cover elements number}
     $|C_v| = \lceil \frac{b}{V \lceil \frac{P}{V} \rceil} \rceil$  {integer flow cover elements number}
     $\lambda^p = \lceil \frac{b}{P} \rceil P - b$  {flow cover excess}
     $\lambda^v = \lceil \frac{b}{V} \rceil V - b$  {integer flow cover excess}
     $\eta = \begin{cases} \lceil \frac{b}{V} \rceil - (|C_v| - 1) \lceil \frac{P}{V} \rceil & \text{if } (b \bmod P) > 0, \\ 0 & \text{otherwise.} \end{cases}$ 
     $\alpha, \beta$  see the proof of proposition 10
  end for
end for

```

- For the case $P \bmod V = 0$, the first $|C_p|$ (respectively $|C_v|$ and $|C_p|$) variables are chosen to construct the flow (respectively integer and mixed) cover set.
- For the case $P \bmod V \neq 0$, the steps of the construction of the flow and the integer flow cover sets are the same as in previous case. However, for the mixed flow cover set, one more step is needed. Since $\beta \geq \alpha$ then, in order to find the inequality the most violated, β must be the coefficient of the biggest valued slack variable j . To find the latter, we proceed as follows. First, we choose $|C_p| - 1$ variables with largest $m()$ value. Then, two values are compared to find the last variable. We choose it either among the first $|C_p|$ variables in table tm or among the variables in $N \setminus C_p$. The two values to be compared are,

1. $m(tm(|C_p|)) - \min_{i \in C_p} \{(\beta - \alpha)y_i\}$ choice among C_p variables,
2. $\max_{i \in N \setminus C_p} \{m(i) - (\beta - \alpha)y_i\}$ choice among $N \setminus C_p$ variables

If (1) is bigger, then the last variable of the mixed cover set is $j = tm(|C_p|)$ and in the MFCI, the smallest $y_i^*, (i \in C_p)$ takes the coefficient β . Otherwise the last variable of the cover set is $j = \operatorname{argmax}_{i \in N \setminus C_p} \{m(i) - (\beta - \alpha)y_i\}$ and in the MFCI, y_j takes the coefficient β .

Once all the cover sets are constructed, the cover inequalities can be written. In table tf , (respect. tI and tm), the first $|C_p|$ (respect. $|C_v|$ and $|C_p|$) variable parts are added to the constant part cf (respect. ci and cm) for the flow (respect. integer and mixed) cover inequalities. The basic cover inequalities are obtained in this manner. Then, for the case $P \bmod V = 0$ or $(P \bmod V \neq 0 \text{ and } \beta \geq \phi)$, to make these inequalities stronger, one has to check the positivity of the variable components for the non-cover elements. If they are positive, then they are added to the LHS of the inequalities, otherwise we stop. If the LHS value obtained is greater than the rhs value $b + s^*(l)$, then the inequality is violated, otherwise there is no violation. The violated inequalities are added to the formulation. This procedure can be applied to the top node or to the middle nodes in the search tree. For the case $P \bmod V = 0$, the implementation of Algorithm 4 requires $O(n^2)$ computations for all the intervals and in each interval the sorting procedure takes $O(n \log(n))$ time. In total, the separation algorithm has a $O(n^3 \log(n))$ time complexity. For the case $P \bmod V \neq 0$, the choice of the slack variable having the coefficient β takes $O(n)$ time but is dominated by $O(n \log(n))$ quick sort algorithm. In total, the separation algorithm takes again $O(n^3 \log(n))$ time.

For the more general case in which P_i and V_i take time varying values, the integer flow cover

Algorithm 3 Computation of the constant and variable components in the LHS of each cover inequality and sorting of the variable parts, $(P, V \text{ constant})$

Require: $n, P, V, |C_p|, |C_v|, \lambda^p, \lambda^v, \eta, \alpha, \beta, (x^*, y^*, z^*, s^*)$

Ensure: Constant and variable parts for each cover inequality

for all t such that $1 \leq t \leq n$ **do**
 for all l such that $t \leq l \leq n$ **do**

 { *Constant part computation* }

$cf \leftarrow (P - \lambda^p)^+ |C_p|$ { *for flow cover* }

$ci \leftarrow (V - \lambda^v)\eta + (V - \lambda^v)\lceil \frac{P}{V} \rceil (|C_v| - 1)$ { *for integer flow cover* }

$cm \leftarrow (V - \lambda^v)\eta + (V - \lambda^v)\lceil \frac{P}{V} \rceil (|C_v| - 1) + \beta + \alpha(|C_p| - 1)$ { *for mixed flow cover* }

 { *Variable part computation* }

for all i such that $t \leq i \leq l$ **do**

$f(i) \leftarrow x_i^* - (P - \lambda^p)^+ y_i^*$ { *Flow cover* }

$I(i) \leftarrow x_i^* - (V - \lambda^v) z_i^*$ { *Integer flow cover* }

$m(i) \leftarrow x_i^* - (V - \lambda^v) z_i^* - \alpha y_i^*$ { *Mixed flow cover* }

end for

 { *Sorting $f(i), I(i)$ and $m(i)$ values in a decreasing order in tables tf, tI and tm respectively.* }

end for

end for

inequality to be lifted and the values $C_p, C_v, \lambda^v, \eta_k$ depend on the choice of the variable k . There are many possibilities to obtain an initial IFCI. Once the integer set is known, which can be done in linear time, the α_i values are computed for each variable in the cover. If we assume $\forall j \in C_v, \sum_{i \in C_v \setminus j} P_i < b$, the coefficient values can be computed analytically, otherwise an NP-hard optimization problem must be solved. For the restrictive case, the separation algorithm takes $O(n^3)$ time. But one can not predict if the inequality obtained is the most violated or not.

6 Computational experiments

In this section, we describe some computational results in order to compare the performance of different formulations and cutting planes for solving LS-SW.

Settings All experiments are carried out on an Intel Xeon 3.2 GHz (bi-processor HT), 4 Gb RAM. For the tests we used Xpress-Mosel 1.6.0. We evaluate the effectiveness of the following formulations and cutting planes:

- The natural formulation **AGG** (see Section 2) solved by a B&B procedure
- Extended formulations **WW-P**, **WW-V**, **DM** (see Section 3.2) solved by B&B (in the tests the two formulations **WW-P** and **WW-V** are coupled for their increasing performance, and denoted by **Both** in the tables)
- Cutting plane using lifted flow (**L-Flow**), lifted integer flow (**L-Int**) and lifted mixed flow (**L-Mix**) cover inequalities (see Sections 4.2 and 4.3). Cutting plane using all the three flow cover inequalities (**All-Cover**). We also give results for the mixed flow cover inequalities without lifting (**Mix**) for some sets.

They are evaluated with the following criteria,

- Initial lower bound values (**LP**: initial linear relaxation value, **SEP**: linear relaxation value obtained after the cutting plane algorithm applied to the root node for the cover type inequalities), compared to the optimal solution value (**OPT**). If OPT is not found after a time limit (300s), then the $\text{gap}\% = \frac{\text{Best Integer Solution} - \text{Best Lower Bound}}{\text{Best Integer Solution}}$ is given,
- Number of cuts added: **cuts**, Time in second: **time**, Number of nodes: **nodes**.

In order to measure the impact of the inequalities from Xpress cuts, we perform two sets of experiments:

- a first set without Xpress cuts,
- a second with Xpress cuts

When Xpress Cuts are added in the formulation by default, we add the prefix **X** in the result tables. The heuristic strategy of Xpress is switched off. The lot sizing problem instances created are for the case with P and V constant. All formulations above can thus be used.

The values of the production capacity and of the number of periods are fixed to $P = 80$ and $n = 20$ respectively. The values of the vehicle capacities and of the average demand, denoted by \bar{d} are varied. We tested three demand configurations: the average demand is low ($\bar{d} = 20$) with big variations from one period to another; the average demand is the half of the production capacity ($\bar{d} = 40$) with small variations and finally the average demand is close to the capacity ($\bar{d} = 60$). We also tested the following configuration $P = 40$, $n = 20$, $V = 10$, $\bar{d} = 30$. The number of periods $n = 40$ is also tested for some sets.

The performance of the cover type inequalities depends on the **objective function**. If, in the objective function, the coefficients of each decision variable are close to their value in the cover inequalities, then they can be more effective and they can cut more infeasible regions. For the test results in Tables 2 to 12 except Tables 4, 5, 6 and 7, we used the objective function $\mathbf{F1} = \sum_{i=1}^{n-\lceil \frac{n}{8} \rceil} x_i + \sum_{i=1}^n (V \lceil \frac{n}{2} \rceil z_i + 2P \lceil \frac{n}{2} \rceil y_i)$. This one seems to have the closest coefficients to those of L-Mix. The second objective function $\mathbf{F2} = \sum_{i=1}^n x_i + \sum_{i=1}^n (1000z_i + 5y_i)$ is more advantageous for L-Int and is used in Tables 4 and 5. The third one is $\mathbf{F3} = \sum_{i=1}^n x_i + \sum_{i=1}^n (5z_i + 1000y_i)$, in which the setup cost is very high compared to the unit production and transportation costs. F3 is used in Tables 6 and 7.

Another parameter changed is the maximum value that can take the variable $s(l)$ in the cutting plane algorithm. In the tables below, $s_{max}(l) = 0$. We also tested some positive values for $s_{max}(l)$. The separation algorithm is called in the root node. At the end of this section, we also make some observations for the case when the separation algorithm is called in each node of the B&B tree (B&Cut). In all experiments, if the optimal solution is not yet found after 300 s, the execution is stopped.

Results We provide results for 12 different sets which we mentioned at the footnote of each table. For example in Table 2, the instance chosen is $P = 80$, $n = 20$, $V = 17$, $\bar{d} = 20$. We have separated the tables of results into four columns. In each column we give respectively the results of the natural formulation AGG, of the extended formulations Both and DM, of the cutting plane algorithm in order to add cover cuts (Mix, L-Mix, L-Flow or L-Int) and finally of the cutting plane algorithm which calls all L-Mix, L-Flow and L-Int cover cuts (All-Cover).

For different test results we observe the following points:

- The LP bound of L-Mix is better than that of Mix.

Most of the time, L-Mix takes less time, generates less cuts and is closer to the optimal solution than Mix. One of the few counterexample we found is for the set $[P = 40, n = 20,$

$V = 12, \bar{d} = 30]$, where Mix explores less nodes and runs in less time than L-Mix. Thus, in Tables 4, 5, 6 and 7, test results for Mix are not given.

Compared to AGG, in most cases, L-Mix is very efficient (see Tables 2 to 12) while Mix and L-Flow can be worse than AGG in terms of the execution time, the total nodes or the final gap (see for instance Tables 2, 3 and 12). For low and high demands, L-Mix is more efficient than for the case where demand does not vary with time and is in average half of the capacity. For some scarce instances (i.e. Table 12), L-Mix solves to the optimum at the root node.

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	12400	14050	14050	12400	12400	12400	12400	12400
SEP	-	-	-	12558	12691	13980	12477	14050
OPT/gap	14081	14081	14081	0.14%	14081	14081	14081	14081
cuts	-	-	-	167	129	70	54	81,65,172
time	207s	11s	125s	> 300s	46s	202s	100s	2.5s
nodes	643616	6109	3595	377784	66942	527532	230770	1226
	X-AGG	X-Both	X-DM	X-Mix	X-L-Mix	X-L-Flow	X-L-Int	X-All-Cover
LP	12481	14050	14050	12481	12481	12481	12481	12481
SEP	-	-	-	12681	12691	13980	12501	14064
OPT/gap	14081	14081	14081	14081	14081	14081	14081	14081
cuts	-	-	-	23	26	28	1	37,19,77
time	88s	10s	190s	100s	20s	139s	86s	$\simeq 0$ s
nodes	384542	4361	4990	338365	49821	379738	318688	406

Table 2: $P = 80, n = 20, V = 17, \bar{d} = 20$, objective function F_1

- The LP bound after the addition of L-Flow is significantly improved.

For low demand, the LP value is increased by L-Flow but the total time can also be increased (see Tables 2, 3, 9, etc.). The improvement depends strongly on the cost repartition. In Tables 6 and 7, surprisingly L-Flow decreases the initial gap between the value of the LP relaxation and the best integer value but this time it takes more time and explores more nodes than all the other methods. Except L-Flow, all the other methods are very efficient for this cost configuration F_3 in which the setup cost is very high compared to the other components.

- The increase on the LP value by the addition of L-Int is small but for the instances where $P \bmod V = 0$, the LP value obtained is close to that obtained by L-Mix (see Tables 3, 9 and 12).
- Most of the time, the LP value given by All-Cover, Both and DM are identical. For some instances, the LP value of All-Cover is even better than that of Both and DM (see Tables 8, 10 and 11).

Surprisingly, the addition of all the lifted flow cover inequalities (L-Flow, L-Int and L-Mix) has the same effect than the addition of DM or Both into the formulation. For the sets tested, where DM and Both solve the problem to the optimum at the root node, the addition of All-Cover gives also the optimum on the root node (see Tables 3, 9, 12). For $P \bmod V = 0$, the linear relaxation of DM and Both gives the optimal solution. However when $P \bmod V \neq 0$, DM and Both sometimes can not give directly the optimal solution on the root node. For this case, DM can take more time than L-Mix and All-Cover with a worse final gap (see Tables 2, 8, 10 and 11).

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	12400	14170	14170	12400	12400	12400	12400	12400
SEP	-	-	-	12431	12979	13980	12970	14170
OPT/gap	14170	14170	14170	1.13%	14170	1.21%	14170	14170
cuts	-	-	-	41	51	42	44	46,39,43
time	171s	$\simeq 0$ s	2s	> 300s	4s	> 300s	4s	$\simeq 0$ s
nodes	798349	1	3	752838	10608	644631	13362	1
	X-AGG	X-Both	X-DM	X-Mix	X-L-Mix	X-L-Flow	X-L-Int	X-All-Cover
LP	12431	14170	14170	12431	12431	12431	12431	12431
SEP	-	-	-	12431	12989	13980	12989	14170
OPT/gap	1.13%	14170	14170	1.13%	14170	0.13%	14170	14170
cuts	-	-	-	6	7	26	3	29,15,23
time	> 300s	$\simeq 0$ s	1.5s	> 300s	11s	> 300s	11.5s	$\simeq 0$ s
nodes	650799	1	1	663766	36678	681408	36591	1

Table 3: $P = 80$, $n = 20$, $V = 20$, $\bar{d} = 20$, objective function F_1

	AGG	Both	DM	L-Mix	L-Flow	L-Int	All-Cover
LP	24014	24431	24431	24014	24014	24014	24014
SEP	-	-	-	24021	24019	24426	24431
OPT/gap	1.71%	24431	24431	1.71%	1.71%	24431	24431
cuts	-	-	-	204	80	64	103,72,200
time	> 300s	1s	6s	> 300s	> 300s	23s	1s
nodes	902413	663	317	376461	720335	56885	625

Table 4: $P = 80$, $n = 20$, $V = 17$, $\bar{d} = 20$, objective function F_2

- Concerning the running time of the All-Cover, DM and Both: DM explores always less nodes in the time limits because of the big size of the formulation which implies more time spent in each node. In most cases, Both takes less time than DM and there are less cases where L-Mix can be better than Both in terms of the running time (see Tables 8, 10, etc.). For almost all the sets tested, All-Cover is the most efficient of all the methods. Even for $P \bmod V \neq 0$, the total time and the total number of nodes are decreased significantly compared to the other methods.

	AGG	Both	DM	L-Mix	L-Flow	L-Int	All-Cover
LP	41156	41861	41861	41156	41156	41156	41156
SEP	-	-	-	41857	41161	41857	41861
OPT/gap	1.70%	41861	41861	41861	1.70%	41861	41861
cuts	-	-	-	70	58	42	58,42,55
time	> 300s	$\simeq 0$ s	1s	41s	> 300s	30s	$\simeq 0$ s
nodes	949567	25	3	90295	842711	96057	3

Table 5: $P = 80$, $n = 20$, $V = 20$, $\bar{d} = 20$, objective function F_2

- The addition of different cover type inequalities is most of the time quite advantageous for each objective function tested.

For the different objective function coefficient tested (apart from F1, F2 and F3, we also performed tests with different objective functions, under different capacity and demand configurations), with the addition of cover type cuts, we always observe an improvement of the LP value, of the total number of nodes and of the total time, compared to the non-

addition of these cuts. One of the few counterexample is given in Table 6. The improvement depends strongly on the cost distribution.

For the objective function F_2 , the addition of L-Int decreases very significantly the initial gap between the value of the LP relaxation and the best integer value (from 1.7% to 0.2%). The total time and the number of nodes also decrease (see Tables 4 and 5). All-Cover is again very efficient compared to the other methods.

	AGG	Both	DM	L-Mix	L-Flow	L-Int	All-Cover
LP	5531	6521	6521	5531	5531	5531	5531
SEP	-	-	-	5542	6518	5533	6521
OPT/gap	6521	6521	6521	6521	6521	6521	6521
cuts	-	-	-	107	70	53	76,58,154
time	145s	\simeq 0s	6s	18s	243s	34s	\simeq 0s
nodes	544129	238	75	30984	587348	94486	21

Table 6: $P = 80$, $n = 20$, $V = 17$, $\bar{d} = 20$, objective function F_3

- In most cases, Xpress cuts decrease the total time of the formulation. For some instances, this strategy can also increase the execution time, the final gap and the number of total nodes (see Table 3).

	AGG	Both	DM	L-Mix	L-Flow	L-Int	All-Cover
LP	5513	6506	6506	5513	5513	5513	5513
SEP	-	-	-	5756	6501	5756	6506
OPT/gap	6506	6506	6506	6506	6506	6506	6506
cuts	-	-	-	91	78	52	105,53,104
time	126s	\simeq 0s	3s	9s	268s	5s	\simeq 0s
nodes	790093	1	3	18763	783298	13107	11

Table 7: $P = 80$, $n = 20$, $V = 20$, $\bar{d} = 20$, objective function F_3

- One can not generalize the influence of the value of $s_{max}(l)$ on the running time.

In the cutting plane algorithm, when $s_{max}(l)$ is increased from 0 to the positive values (we tested until $s_{max}(l) = 50$), again it is difficult to generalize the influence of the parameters. For the set chosen in Table 2, when $s_{max}(l) = 50$, the total number of nodes decreases from 66942 to 39675 and the total time from 46s to 33s. 13 more cuts are generated. However, for the set $[P = 40, n = 20, V = 12, \bar{d} = 30]$, the total time increased from 75s to 170s, and the number of nodes explored from 63872 to 74380. 106 more cuts are generated for this case.

- For some sets tested we observe that, when the separation algorithm is called in each node instead of only top node, the number of nodes explored in the limited time decreases but the final gap increases. For larger instances (i.e. $n=40$), the LP time of DM increases. The final gap increases for each formulation. The similar remark above can be made for the larger sets.

7 Conclusions

We identified a new class of valid inequalities for a special case of the lot sizing problem, that we called mixed flow cover inequalities. A single-item is produced with a limited capacity, and

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	24895	26481	26486	24895	24895	24895	24895	24895
SEP	-	-	-	25245	25493	26375	25001	26497
OPT/gap	4.34%	0.60%	0.64%	3.77%	0.48%	0.45%	4.63%	0.32%
cuts	-	-	-	206	161	126	100	149,117,404
time	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s
nodes	609389	343735	24130	254340	557253	554455	619376	131424
	X-AGG	X-Both	X-DM	X-Mix	X-Lift	X-L-Flow	X-L-Int	X-All-Cover
LP	25081	26481	26486	25081	25081	25081	25081	25081
SEP	-	-	-	25409	25493	26389	25082	26543
OPT/gap	0.11%	0.11%	0.64%	0.11%	0.64%	0.11%	0.11%	0.11%
cuts	-	-	-	87	40	45	7	54,20,203
time	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s
nodes	1202272	394473	26675	779255	903474	737250	1177293	796405

Table 8: $P = 80$, $n = 20$, $V = 17$, $\bar{d} = 40$

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	24895	26515	26515	24895	24895	24895	24895	24895
SEP	-	-	-	24906	25326	26375	25315	26515
OPT/gap	0.49%	26515	26515	0.49%	26515	0.49%	26515	26515
cuts	-	-	-	105	97	76	51	74,45,72
time	> 300s	$\simeq 0$ s	1.5s	> 300s	30s	> 300s	20s	$\simeq 0$ s
nodes	1199538	1	1	893000	56163	448135	46406	1
	X-AGG	X-Both	X-DM	X-Mix	X-Lift	X-L-Flow	X-L-Int	X-All-Cover
LP	24906	26515	26515	24906	24906	24906	24906	24906
SEP	-	-	-	24906	25326	26375	25326	26515
OPT/gap	0.49%	26515	26515	0.49%	26515	0.49%	26515	26515
cuts	-	-	-	17	27	27	12	45,17,43
time	> 300s	$\simeq 0$ s	1.5s	> 300s	25s	> 300s	37s	$\simeq 0$ s
nodes	1092156	1	1	1015020	56129	740036	74267	3

Table 9: $P = 80$, $n = 20$, $V = 20$, $\bar{d} = 40$

for each batch produced, a fixed cost is paid in addition to the setup cost per production period. This new class is derived from the integer flow cover inequalities which have been first proposed for the single node flow set which is a generalization of the set X^{mix} we study. For X^{mix} , we provided also the flow and integer flow cover inequalities and different extended formulations we found in the literature. These extended formulations are proposed for X^{mix} itself or for its valid relaxations. We proposed a separation algorithm for the cover type cuts, which runs in polynomial time when the capacities are constant. For X^{mix} , this new class of valid inequalities is shown to be more effective than the flow cover and the integer flow cover inequalities. The insertion of the extended formulations based on the divisible mixing set and on the mixing set are very efficient when the vehicle capacity divides the production capacity. Their linear relaxation gives the optimal solution in this case. Otherwise they can take more time than the violated mixed flow cuts added to the root node. Surprisingly, for the sets tested until now, the addition of all the violated cover type inequalities (lifted flow, integer and mixed flow cover), the total time and the total number of nodes decreased significantly. For the case where the vehicle capacity divides the production capacity, we obtain the optimum at the root node after the addition of all cover type inequalities, as the extended formulations do. Even for the more general case where the vehicle capacity does not divide the production capacity, we obtain very good results compared to other

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	38415	39239	39286	38415	38415	38415	38415	38415
SEP	-	-	-	40002	40037	39135	38558	40117
OPT/gap	42440	42440	2.76%	42440	42440	42440	42440	42440
cuts	-	-	-	190	157	42	53	44,53,210
time	67s	70s	> 300s	7s	3.8s	66s	50s	3s
nodes	130428	70687	29185	3499	3010	124973	80234	1227
	X-AGG	X-Both	X-DM	X-Mix	X-Lift	X-L-Flow	X-L-Int	X-All-Cover
LP	42046	40009	42130	42046	42046	42046	42046	42046
SEP	-	-	-	42046	42046	42046	42130	42130
OPT/gap	42440	42440	42440	42440	42440	42440	42440	42440
cuts	-	-	-	3	3	0	16	0,16,3
time	$\simeq 0$ s	2.5s	9s	$\simeq 0$ s	$\simeq 0$ s	$\simeq 0$ s	$\simeq 0$ s	$\simeq 0$ s
nodes	333	2321	506	323	323	333	284	284

Table 10: $P = 80$, $n = 20$, $V = 70$, $\bar{d} = 60$

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	24895	26736	26736	24895	24895	24895	24895	24895
SEP	-	-	-	26802	26698	26386	25256	26875
OPT/gap	4.50%	4.41%	5.71%	2.75%	2.19%	2.72%	4.77%	3.83%
cuts	-	-	-	356	170	66	58	80,86,1726
time	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s	> 300s
nodes	659350	300315	24202	105330	295192	560897	653341	14995
	X-AGG	X-Both	X-DM	X-Mix	X-Lift	X-L-Flow	X-L-Int	X-All-Cover
LP	27218	26736	26736	27218	27218	27218	27218	27218
SEP	-	-	-	27218	27218	27218	27423	27423
OPT/gap	28336	2.65%	4.96%	28336	28336	28336	28336	28336
cuts	-	-	-	11	13	9	24	8,23,12
time	8s	> 300s	> 300s	11s	10s	9s	24s	25s
nodes	12207	253582	24733	19355	22620	18193	47443	56523

Table 11: $P = 80$, $n = 20$, $V = 70$, $\bar{d} = 40$

methods, in terms of the total time, the final gap and the number of nodes. The efficiency of each method depends on the cost repartition.

We established different cases where the cover inequalities and extended formulations are and are not efficient. We performed many experiments to evaluate the performance of all the optimization methods, comparing the total execution time, the total number of nodes explored and the linear relaxation value at the top node. The parameters which influence these criteria are found to be the coefficients of the objective function, the demand repartition and its closeness to the production capacity, the vehicle capacities, the number of periods and the value of the stock variable in the cutting plane. More investigation on the case where the production and the vehicle capacities take time varying values deserves attention. Preliminary observations show that in this case, with the addition of the mixed flow cover inequalities, a gain is possible in terms of the total execution time and the final gap. A new separation algorithm has to be proposed for the more general case, solving an NP-hard optimization problem.

Acknowledgement We are grateful to L.A. Wolsey for his careful reading of this paper.

	AGG	Both	DM	Mix	L-Mix	L-Flow	L-Int	All-Cover
LP	18415	18556	18556	18415	18415	18415	18415	18415
SEP	-	-	-	18436	18556	18516	18536	18556
OPT/gap	18556	18556	18556	18556	18556	18556	18556	18556
cuts	-	-	-	17	10	10	6	10,6,9
time	3s	$\simeq 0$ s	0.5s	4s	$\simeq 0$ s	2s	$\simeq 0$ s	$\simeq 0$ s
nodes	11277	1	3	14351	1	7133	5	1
	X-AGG	X-Both	X-DM	X-Mix	X-Lift	X-L-Flow	X-L-Int	X-All-Cover
LP	18436	18556	18556	18436	18436	18436	18436	18436
SEP	-	-	-	18436	18556	18516	18556	18556
OPT/gap	18556	18556	18556	18556	18556	18556	18556	18556
cuts	-	-	-	7	5	4	3	4,3,4
time	4s	$\simeq 0$ s	0.7s	3.3s	$\simeq 0$ s	3.2s	$\simeq 0$ s	$\simeq 0$ s
nodes	14565	1	1	11257	1	8045	1	1

Table 12: $P = 40$, $n = 20$, $V = 10$, $\bar{d} = 30$

References

- [1] A. Akbalık and C. Rapine. Polynomial time algorithms for constant capacitated lot sizing problems with equal length step-wise linear costs. Technical report, G-SCOP, Institut National Polytechnique de Grenoble, 2006.
- [2] A. Atamtürk. On the facets of the mixed-integer knapsack polyhedron. *Mathematical Programming*, 98:145–175, 2003.
- [3] N. Brahimi, S. Dauzère-Pérès, N.M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168(1):1–16, 2006.
- [4] L.M.A. Chan, A. Muriel, Z.-J. Shen, and D. Simchi-Levi. On the effectiveness of zero-inventory-ordering policies for the economic lot-sizing model with a class of piecewise linear cost structures. *Operations Research*, 50(6):1058–1067, 2002.
- [5] L.M.A. Chan, A. Muriel, and D. Simchi-Levi. Production/distribution planning problems with piecewise linear concave costs. Technical report, University of Toronto, University of Massachusetts and Northwestern University, 2000.
- [6] H-D. Chen, D.W. Hearn, and C-Y. Lee. A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *Journal of Global Optimization*, 4:285–300, 1994.
- [7] M. Diaby and A. Martel. Dynamic lot sizing for multi-echelon distribution systems with purchasing and transportation price discounts. *Operations Research*, 41(1):48–59, 1993.
- [8] M. Florian, J.K. Lenstra, and A.H.G. Rinnooy Kan. Deterministic production planning: Algorithms and complexity. *Management Science*, 26(7):669–679, 1980.
- [9] O. Günlük and Y. Pochet. Mixing mixed integer inequalities. *Mathematical Programming*, (90):429–457, 2001.
- [10] Y. Jin and A. Muriel. Single-warehouse multi-retailer inventory systems with full truck-load shipments. Technical report, Department of Mechanical and Industrial Engineering, University of Massachusetts, 2005.

- [11] C.-Y. Lee, S. Çetinkaya, and W. Jaruphongsa. A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse. *Operations Research*, 51(5):735–747, 2003.
- [12] C.-L. Li, V.N. Hsu, and W.-Q. Xiao. Dynamic lot sizing with batch ordering and truckload discounts. *Operations Research*, 52(4):639–654, 2004.
- [13] S.A. Lippman. Optimal inventory policy with multiple set-up costs. *Management Science*, 16(1):118–138, 1969.
- [14] Q. Louveaux and L.A. Wolsey. Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. *4OR*, 1(3):173–207, 2003.
- [15] M.W. Padberg, T.J. van Roy, and L.A. Wolsey. Valid linear inequalities for fixed charge problems. *Operations Research*, 33:842–861, 1985.
- [16] Y. Pochet and L.A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer, 2006.
- [17] Y. Pochet and L.A. Wolsey. Lot-sizing with constant batches: Formulation and valid inequalities. *Mathematics of Operations Research*, 18(4):767–785, 1993.
- [18] Y. Pochet and L.A. Wolsey. Polyhedra for lot sizing with Wagner-Whitin costs. *Mathematical Programming*, 67:297–324, 1994.
- [19] D.X. Shaw and A.P.M. Wagelmans. An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. *Management Science*, 44(6):831–838, 1998.
- [20] C. Swoveland. A deterministic multi-period production planning model with piecewise concave production and holding-backorder costs. *Management Science*, 21(9):1007–1013, 1975.
- [21] M. Van Vyve. *A solution approach of production planning problems based on compact formulations for single-item lot-sizing models*. PhD thesis, UCLouvain, 2003.

Proof for the trivial facets of X^{mix}

- [illegible]

6. $\sum_{i \in N} x_i \leq b$, for $|C_p| < n$

At least one of the variables can take 0 value and the others are saturated (n points are obtainable in this manner). For the other n points, the 0 valued y and z variables can be increased by one. For the last n points, the 0 valued y and z variables can be increased by 1 and $\lceil \frac{P}{V} \rceil$ units respectively. For the final point, for any of these $3n$ points, the variable s value can be increased by one unit.

- $x_i \leq ay_i + bz_i, \forall i \in N$
 $\forall i \in N \begin{cases} x_i \leftarrow P & y_i \leftarrow 1 & z_i \leftarrow \lceil \frac{P}{V} \rceil & s \leftarrow 0 \\ x_j \leftarrow 0 & y_j \leftarrow 0 & z_j \leftarrow 0 & s \leftarrow 0 \end{cases} \quad \forall j \in N \setminus i$
 $\forall i \in N \begin{cases} x_i \leftarrow \lfloor \frac{P}{V} \rfloor V & y_i \leftarrow 1 & z_i \leftarrow \lfloor \frac{P}{V} \rfloor & s \leftarrow 0 \\ x_j \leftarrow 0 & y_j \leftarrow 0 & z_j \leftarrow 0 & s \leftarrow 0 \end{cases} \quad \forall j \in N \setminus i$
 $\forall i \in N \begin{cases} x_i \leftarrow \lfloor \frac{P}{V} \rfloor V & y_i \leftarrow 1 & z_i \leftarrow \lfloor \frac{P}{V} \rfloor & s \leftarrow 0 \\ x_j \leftarrow 0 & y_j \leftarrow 1 & z_j \leftarrow 1 & s \leftarrow 0 \end{cases} \quad \forall j \in N \setminus i$
 The last point ; $\{ x_j \leftarrow 0 \quad y_j \leftarrow 0 \quad z_j \leftarrow 0 \quad s \leftarrow 1 \quad \forall j \in N$

Appendix B1

1. $3|C_v|$ linearly independent directions for the case $n = |C_v|$. Other assumptions are $P \bmod V = 0$, $b \bmod V \neq 0$ and $L = \emptyset$. \vec{e}_t is the unit vector $\forall t \in \{1, \dots, 3n+1\}$. We denote by $\vec{e}_t' = \vec{e}_{n+t}$ and by $\vec{e}_t'' = \vec{e}_{2n+t}$. X_t, Y_t, Z_t and S_t describes the vectors.

Direction (F-S)	First solution (F)	Second solution (S)
$\forall t \in [2, C_v]$	$\forall i \in [2, C_v]$	$\forall i \in [1, C_v] \setminus t$
$X_t = \lambda^p \vec{e}_t, Z_t = (\frac{P}{V} - \eta) \vec{e}_t''$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$
$X_1 = -\lambda^p \vec{e}_1, Z_1 = (\eta - \frac{P}{V}) \vec{e}_1''$	$x_1 = P - \lambda^p, y_1 = 1, z_1 = \eta$	$x_t = P - \lambda^p, y_t = 1, z_t = \eta$
$\forall t \in C_v \quad X_t = (P - \lambda^p) \vec{e}_t$	$\forall i \in C_v \setminus t$	$\forall i \in C_v \setminus t$
$Y_t = \vec{e}_t', Z_t = \eta \vec{e}_t''$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$
	$x_t = P - \lambda^p, y_t = 1, z_t = \eta$	$x_t = 0, y_t = 0, z_t = 0$
$\text{if } \eta > 1$	$\forall i \in C_v \setminus t$	$\forall i \in C_v \setminus t$
$\forall t \in C_v$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$
$X_t = (V - \lambda^v) \vec{e}_t, Z_t = \vec{e}_t''$	$x_t = P - \lambda^p, y_t = 1, z_t = \eta$	$x_t = (\eta - 1)V, y_t = 1, z_t = \eta - 1$
$\text{if } \eta = 1$	$\forall i \in C_v \setminus t$	$x_j = P - V$
$\forall t \in C_v, \exists j \in C_v \setminus t$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$	$x_t = V, y_t = 1, z_t = 1$
$X_t = -\lambda^v \vec{e}_t, Z_t = (\eta - 1) \vec{e}_t''$	$x_t = P - \lambda^p, y_t = 1, z_t = \eta$	$\forall i \in C_v \setminus \{t, j\}$
$X_j = V \vec{e}_j, Z_j = \vec{e}_j''$		$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$
$\forall i \in C_v$	$\forall i \in C_v$	$\forall i \in [2, C_v]$
$X_1 = \lambda^p \vec{e}_1, S = \lambda^p \vec{e}_{3n+1}$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = \lambda^p$	$x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$
$Z_1 = (\frac{P}{V} - \eta) \vec{e}_1''$		$x_1 = P - \lambda^p, y_1 = 1, z_1 = \eta$

Table 13: $3|C_v|$ linearly independent directions for the case $n = |C_v|$.

2. $3|C_v|$ linearly independent directions are given in Table 13. One has to find another $3(n - |C_v|)$ direction to show that inequality (15) is still facet defining for $n > |C_v|$. In Table 14 we give only these additional $3(n - |C_v|)$ directions. The cover elements indices are ordered in the interval $[1, |C_v|]$. The other assumptions remain the same.

Direction (S-F)	First solution (F)	Second solution (S)
$\forall t \in N \setminus C_v$ $X_t = (P - \lambda^p) \vec{e}_t, Y_t = \vec{e}_t', Z_t = \eta \vec{e}_t''$ $X_{ C_v } = (P - \lambda^p) \vec{e}_{ C_v }$ $Y_{ C_v } = \vec{e}_{ C_v }', Z_{ C_v } = \eta \vec{e}_{ C_v }''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $\forall j \in N \setminus C_v$ $x_j = 0, y_j = 0, z_j = 0$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = 0, y_{ C_v } = 0, z_{ C_v } = 0$ $x_t = P - \lambda^p, y_t = 1, z_t = \eta$ $\forall j \in (N \setminus C_v) \setminus t, x_j = y_j = z_j = 0$
$\forall t \in (N \setminus C_v)$ $Y_t = \vec{e}_t', Z_t = \vec{e}_t''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $\forall j \in (N \setminus C_v)$ $x_j = 0, y_j = 0, z_j = 0$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $x_t = 0, y_t = 1, z_t = 1$ $\forall j \in (N \setminus C_v) \setminus t, x_j = y_j = z_j = 0$
$\forall t \in (N \setminus C_v)$ $Z_t = (\frac{P}{V} - 1) \vec{e}_t''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $x_t = 0, y_t = 1, z_t = 1$ $\forall j \in (N \setminus C_v) \setminus t, x_j = y_j = z_j = 0$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $x_t = 0, y_t = 1, z_t = \frac{P}{V}$ $\forall j \in (N \setminus C_v) \setminus t, x_j = y_j = z_j = 0$

Table 14: $3(n - |C_v|)$ linearly independent directions for the case $n = |C_v|$.

3. See Table 15 for $3|L|$ linear directions.

Direction	First solution	Second solution
$\forall t \in L$ $X_t = (P - \lambda^p) \vec{e}_t, Y_t = \vec{e}_t', Z_t = \eta \vec{e}_t''$ $X_{ C_v } = -(P - \lambda^p) \vec{e}_{ C_v }$ $Y_{ C_v } = -\vec{e}_{ C_v }', Z_{ C_v } = -\eta \vec{e}_{ C_v }''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = P - \lambda^p, y_{ C_v } = 1, z_{ C_v } = \eta$ $\forall j \in [C_v + 1, n]$ $x_j = 0, y_j = 0, z_j = 0$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = 0, y_{ C_v } = 0, z_{ C_v } = 0$ $x_t = P - \lambda^p, y_t = 1, z_t = \eta$ $\forall j \in [C_v + 1, n] \setminus t, x_j = y_j = z_j = 0$
$\forall t \in L$ $X_t = P \vec{e}_t, Y_t = \vec{e}_t', Z_t = \frac{P}{V} \vec{e}_t''$ $X_{ C_v } = -P \vec{e}_{ C_v }, Y_{ C_v } = -\vec{e}_{ C_v }'$ $Z_{ C_v } = -\frac{P}{V} \vec{e}_{ C_v }''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v } = y_{ C_v } = z_{ C_v } = 0$ $\forall j \in [C_v + 1, n]$ $x_j = 0, y_j = 0, z_j = 0$	$\forall i \in [1, C_v - 2]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $x_{ C_v -1} = y_{ C_v -1} = z_{ C_v -1} = 0$ $x_{ C_v } = y_{ C_v } = z_{ C_v } = 0$ $x_t = P, y_t = 1, z_t = \frac{P}{V}$ $\forall j \in [C_v + 1, n] \setminus t, x_j = y_j = z_j = 0$
<u>if $\eta > 1$</u> $\forall t \in L, X_t = (\eta - 1) V \vec{e}_t$ $Y_t = \vec{e}_t', Z_t = (\eta - 1) \vec{e}_t''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $\forall j \in [C_v , n]$ $x_j = y_j = z_j = 0$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $\forall j \in [C_v , n] \setminus t$ $x_j = y_j = z_j = 0$ $x_t = (\eta - 1) V, y_t = 1, z_t = \eta - 1$
<u>if $\eta = 1$</u> $\forall t \in L, X_t = V \vec{e}_t, Y_t = \vec{e}_t', Z_t = \vec{e}_t''$ $X_{ C_v -1} = -V \vec{e}_{ C_v -1}$ $Z_{ C_v -1} = -\vec{e}_{ C_v -1}''$	$\forall i \in [1, C_v - 1]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $\forall j \in [C_v , n]$ $x_j = y_j = z_j = 0$	$\forall i \in [1, C_v - 2]$ $x_i = P, y_i = 1, z_i = \frac{P}{V}, s = 0$ $\forall j \in [C_v , n] \setminus t, x_j = y_j = z_j = 0$ $x_{ C_v -1} = P - V, y_{ C_v -1} = 1$ $z_{ C_v -1} = \lceil \frac{P-V}{V} \rceil$ $x_t = V, y_t = 1, z_t = 1$

Table 15: $3|L|$ linearly independent directions.

Appendix C1

Some abbreviations are, T.F: trivial facets (see proposition 5), F.C: flow cover inequality, L.F.C: lifted FC, I.C: integer flow cover inequality, L.I.C: lifted IC, M.C: mixed flow cover inequality, L.M.C: lifted MC, Nmb: total number of inequalities. In the column *Others*, the number of inequalities that could not be classified as cover inequality nor trivial facets is given.

b	T.F.	F.C.	L.F.C.	I.C.	L.I.C.	M.C	L.M.C.	Others	Nmb.
≥ 27	15	-	-	-	-	-	-	-	15
26, 25	15	-	-	-	-	1	-	-	16
24	15	1	-	-	-	-	-	-	16
23, 22	15	1	-	-	-	1	-	-	17
21	15	1	-	-	-	-	-	-	16
20, 19	15	1	-	1	-	-	-	3	20
18	15	1	-	-	-	-	-	3	19
17, 16	16	-	-	-	1	3	1	12	33
15	16	3	1	-	-	-	-	3	23
14, 13	16	3	1	-	1	3	1	15	40
12	16	3	1	-	-	-	-	3	23
11, 10	16	3	1	3	1	-	-	18	45
9	16	-	-	-	-	-	-	12	28
8, 7	16	-	-	-	4	3	-	66	89
6	16	3	-	-	-	-	-	9	28
5, 4	13	3	-	-	4	3	-	12	35
≤ 3	13	3	-	-	-	-	-	-	16

Table 16: Number of different facet defining inequalities for X^{mix} in the case where $P \bmod V = 0$: complete list for each b value in \mathbb{Z}_+ . ($P = 9, V = 3, n = 3$)

Appendix C2

b	T.F.	F.C.	L.F.C.	I.C.	L.I.C.	M.C	L.M.C.	Others	Nmb.
≥ 27	18	-	-	-	-	-	-	-	18
26	18	1	-	-	-	1	-	-	20
25	18	1	-	-	-	-	-	6	25
24	18	1	-	-	-	-	-	18	37
23	18	1	-	-	-	-	1	22	42
22	18	1	-	-	-	-	1	13	33
21	18	1	-	-	-	-	1	12	32
20	18	1	-	-	-	-	-	31	50
17	19	3	1	-	1	-	-	21	45
15	19	3	1	-	1	3	1	145	173
14	19	3	1	-	1	3	1	118	146
13	19	3	1	-	1	3	1	30	58
10	19	3	1	-	4	-	-	130	157
8	16	3	-	-	-	-	-	9	28
≤ 4	16	-	-	-	-	-	-	-	16

Table 17: Number of different facet defining inequalities for X^{mix} in the case where $P \bmod V \neq 0$. ($P = 9, V = 4, n = 3$)

Appendix D

All facet defining inequalities for $b = 14, P = 9, V = 3$ are given below. $|C_p| = 2, |C_v| = 2, \lambda^p = 4, \lambda^v = 1, \eta = 2$.

- Trivial facet defining inequalities (16 in total);

$$\begin{aligned}
& - x_i \geq 0, \forall i \in \{1, \dots, 3\} \\
& - y_i \leq z_i, \forall i \in \{1, \dots, 3\} \\
& - z_i \leq 3y_i, \forall i \in \{1, \dots, 3\} \\
& - x_i \leq 3z_i, \forall i \in \{1, \dots, 3\} \\
& - y_i \leq 1, \forall i \in \{1, \dots, 3\} \\
& - \sum_{i \in \{1, \dots, 3\}} x_i \leq 14
\end{aligned}$$

- Flow cover inequalities;

$$\begin{aligned}
x_2 + x_3 - 5y_2 - 5y_3 &\leq 4 \\
x_1 + x_3 - 5y_1 - 5y_3 &\leq 4 \\
x_1 + x_2 - 5y_1 - 5y_2 &\leq 4
\end{aligned}$$

- Lifted flow cover inequalities;

$$x_1 + x_2 + x_3 - 5y_1 - 5y_2 - 5y_3 \leq 4$$

- Lifted integer flow cover inequalities;

$$x_1 + x_2 + x_3 - 2z_1 - 2z_2 - 2z_3 \leq 4$$

- Mixed flow cover inequalities;

$$\begin{aligned}
x_2 + x_3 - y_2 - y_3 - 2z_2 - 2z_3 &\leq 2 \\
x_1 + x_3 - y_1 - y_3 - 2z_1 - 2z_3 &\leq 2 \\
x_1 + x_2 - y_1 - y_2 - 2z_1 - 2z_2 &\leq 2
\end{aligned}$$

- Lifted mixed flow cover inequalities;

$$x_1 + x_2 + x_3 - y_1 - y_2 - y_3 - 2z_1 - 2z_2 - 2z_3 \leq 2$$

- Others

$$\begin{aligned}
x_1 + x_2 + x_3 - 2y_1 + y_2 + y_3 - 2z_1 - 3z_2 - 3z_3 &\leq 2 \\
x_1 + x_2 + x_3 + y_1 - 2y_2 + y_3 - 3z_1 - 2z_2 - 3z_3 &\leq 2 \\
x_1 + x_2 + x_3 + y_1 + y_2 - 2y_3 - 3z_1 - 3z_2 - 2z_3 &\leq 2 \\
x_1 + x_2 + x_3 - 5y_1 - 5y_2 - y_3 - 2z_3 &\leq 4 \\
x_1 + x_2 + x_3 - 5y_1 - y_2 - 5y_3 - 2z_2 &\leq 4 \\
x_1 + x_2 + x_3 - y_1 - 5y_2 - 5y_3 - 2z_1 &\leq 4 \\
x_1 + 3x_2 + 3x_3 - 5y_1 - 3y_2 - 3y_3 - 6z_2 - 6z_3 &\leq 6 \\
3x_1 + x_2 + 3x_3 - 3y_1 - 5y_2 - 3y_3 - 6z_1 - 6z_3 &\leq 6 \\
3x_1 + 3x_2 + x_3 - 3y_1 - 3y_2 - 5y_3 - 6z_1 - 6z_2 &\leq 6 \\
x_1 + x_2 + x_3 - 2y_1 - 2y_2 + 2y_3 - 2z_3 &\leq 10 \\
x_1 + x_2 + x_3 - 2y_1 + 2y_2 - 2y_3 - 2z_2 &\leq 10 \\
x_1 + x_2 + x_3 + 2y_1 - 2y_2 - 2y_3 - 2z_1 &\leq 10 \\
x_1 + 3x_2 + 3x_3 - 2y_1 - 6z_2 - 6z_3 &\leq 12 \\
3x_1 + x_2 + 3x_3 - 2y_2 - 6z_1 - 6z_3 &\leq 12 \\
3x_1 + 3x_2 + x_3 - 2y_3 - 6z_1 - 6z_2 &\leq 12
\end{aligned}$$

Algorithm 4 Separation algorithm for the cover inequalities

Require: n, P, V, b and all values returned by Algorithm 2 and 3

Ensure: Violated inequalities *flow*, *integer*, *mixed* or no violation

IF(condition, a, b) returns a if condition is true, otherwise returns b .

for all k such that $1 \leq k \leq n$ **do**

for all l such that $k \leq l \leq n$ **do**

 Call Algorithm 3

 { *Violation of flow cover inequality* }

$violf \leftarrow cf + \sum_{j \in C_p} f(tf(j)) + \sum_{j \in N \setminus C_p} IF(f(tf(j)) > 0, f(tf(j)), 0)$

if ($violf > b + s^*(l)$) **then**

$flow \leftarrow cf + \sum_{j \in C_p} (x_{tf(j)} - (P - \lambda^p)y_{tf(j)}) + \sum_{j \in N \setminus C_p} IF(f(tf(j)) > 0, x_{tf(j)} - (P - \lambda^p)y_{tf(j)}, 0)$

else

 Return non violation of flow cover inequality

end if

 { *Violation of integer flow cover inequality* }

$viol_i \leftarrow ci + \sum_{j \in C_v} I(ti(j)) + \sum_{j \in N \setminus C_v} IF(I(ti(j)) > 0, I(ti(j)), 0)$

if ($viol_i > b + s^*(l)$) **then**

$integer \leftarrow ci + \sum_{j \in C_v} (x_{ti(j)} - (V - \lambda^v)z_{ti(j)}) + \sum_{j \in N \setminus C_v} IF(I(ti(j)) > 0, x_{ti(j)} - (V - \lambda^v)z_{ti(j)}, 0)$

else

 Return non violation of integer flow cover inequality

end if

 { *Violation of mixed flow cover inequality* }

$violm \leftarrow cm + \sum_{j \in C_v \cup S} m(tm(j)) + \sum_{j \in N \setminus (C_v \cup S)} IF(m(tm(j)) > 0, m(tm(j)), 0)$

if ($violm > b + s^*(l)$) **then**

$mixed \leftarrow cm + \sum_{j \in C_v \cup S} (x_{tm(j)} - (V - \lambda^v)z_{tm(j)} - \alpha y_{tm(j)}) + \sum_{j \in N \setminus (C_v \cup S)} IF(m(tm(j)) > 0, x_{tm(j)} - (V - \lambda^v)z_{tm(j)} - \alpha y_{tm(j)}, 0)$

else

 Return non violation of mixed flow cover inequality

end if

end for

end for
